2

Systems
Optimization
Laboratory

DTIC FILE COPY

AD-A228 829

[redacted]

DTIC
ELECTE
NOV 15 1990
S B D

Department of Operations Research
Stanford University
Stanford, CA 94305

90 11 13 135

SYSTEMS OPTIMIZATION LABORATORY
DEPARTMENT OF OPERATIONS RESEARCH
STANFORD UNIVERSITY
STANFORD, CALIFORNIA 94305-4022

A Sequential Quadratic Programming Algorithm Using an Incomplete Solution of the Subproblem

by
Walter Murray* and Francisco J. Prieto†

TECHNICAL REPORT SOL 90-12

September 1990

* Research and reproduction of this report were partially supported by the National Science Foundation Grant DDM-8715153 and the Office of Naval Research Grant N00014-90-J-1242.

† Research partially supported by a grant from the Bank of Spain.

A SEQUENTIAL QUADRATIC PROGRAMMING ALGORITHM USING AN INCOMPLETE SOLUTION OF THE SUBPROBLEM

WALTER MURRAY* AND FRANCISCO J. PRIETO†

*Systems Optimization Laboratory
Department of Operations Research
Stanford University

†Dept. de Automática, Ingeniería Electrónica e Informática Industrial
E.T.S. Ingenieros Industriales
Universidad Politécnica, Madrid

Technical Report SOL 90-12
September 1990

Abstract

A feature of current sequential quadratic programming (SQP) methods to solve nonlinear constrained optimization problems is the necessity at each iteration to solve a quadratic program (QP). We show that if the QP subproblem is convex and an active-set method is used to solve it, then there exist iterates other than the minimizer that may be used to define a suitable search direction. None of the usual properties of an SQP method are compromised by the new definition of the search direction.

We derive some new properties for an SQP method that uses a particular augmented Lagrangian merit function. Specifically we show, under suitable additional assumptions, that the rate of convergence is superlinear. We also show that the penalty parameter used in the merit function is bounded.

1. Introduction

The problem of interest is the following:

$$\boxed{\begin{array}{ll} \underset{x\in\Re^n}{\text{minimize}} & F(x) \\ \text{s.t.} & c(x) \geq 0, \end{array}} \qquad \text{NP}$$

*Research supported by the National Science Foundation Grant DDM-8715153 and the Office of Naval Research Grant N00014-90-J-1242.

†Research partially supported by a grant from the Bank of Spain.

where $F : \Re^n \to \Re$ and $c : \Re^n \to \Re^m$. More specifically, we are interested in computing x^*, a *first-order KKT point* of NP. Such points are feasible and satisfy the following conditions:

$$\nabla F(x^*) = \nabla c(x^*)^T \lambda^*, \qquad \lambda_i^* c_i(x^*) = 0 \quad i = 1, \ldots, m$$

for some nonnegative multiplier vector $\lambda^* \in \Re^m$. Since we shall not assume second derivatives are known, this is the best that can be achieved, and so whenever the term "solution point" is used in the following sections, what will be meant is a first-order KKT point of NP.

We study the convergence properties of a sequential quadratic programming (SQP) algorithm in which the quadratic programming subproblems generated are convex. The unique feature of the algorithm studied is that the search direction in a given iteration is computed as an "incomplete solution" of the quadratic subproblem. A precise definition of the term "incomplete solution" will be given in Section 2.

Typically SQP algorithms generate a sequence of points $\{x_k\}$ converging to a solution, by solving at each point, x_k, a quadratic program (QP) of the form

$$\boxed{\begin{array}{ll} \underset{p \in \Re^n}{\text{minimize}} & \nabla F(x_k)^T p + \frac{1}{2} p^T H_k p \\ \text{s.t} & c(x_k) + \nabla c(x_k) p \geq 0 \end{array}} \qquad \text{QP}$$

for some positive definite matrix H_k. Let p_k (referred to as the search direction) denote the solution to QP. The next point in the sequence is defined to be the result of a linesearch from x_k along p_k, in such a way that the value of a certain merit function is decreased.

A number of different merit functions have been proposed. The one studied in this paper is that analysed in [GMSW86b] and used in the algorithm NPSQP. It is similar to merit functions proposed by Wright [Wri76] and Schittkowski [Sch81].

Although our primary interest is this specific merit function, we also show (Section 5) how the ideas discussed can be extended to the use of other merit functions.

SQP algorithms in general and NPSQP in particular are viewed by many as the best approach to the solution of NP when n is small (< 200). As the size of the problem grows, usually so does the relative importance of the effort to solve QP when compared to the total effort. Indeed for many large problems the effort to solve QP dominates the total effort.

Because the unique minimizer of QP is used to define the search direction, it is not necessary in any theoretical discussion of an SQP algorithm to define *how* the QP subproblem is solved. In practice by far the commonest techniques used to solve QPs are active-set methods. The implementation of NPSQP called NPSOL uses such a method. For a comprehensive survey of active-set methods see [GMW81], [Fle87] and [GMSW88]. When an active-set method is used the potential number of iterations to solve a QP grows exponentially with n. In practice the number of iterations grows much more slowly than exponential (if this was not the case active-set methods would be hopelessly inefficient). Nonetheless, the number of iterations required to solve a large QP is usually large. In any implementation of an SQP method it is necessary to limit the number of iterations allowed to solve a given QP subproblem. If the QP solution process is terminated prematurely the

SQP algorithm may break down. It is in part for this reason that the development of SQP methods for large-scale problems has been inhibited. Even for small problems there are occasions when the number of QP iterations are excessive. Since the definition of "small" continues to increase as computers become more powerful we can expect the cost of solving the subproblems to grow in importance.

Typically the number of iterations to solve the QP subproblem is large in the initial iterations and falls exponentially. Near the solution only a single iteration is usually required. A natural question is whether the effort to solve the QP subproblem is warranted far from the solution when the information used to construct the subproblem may be unreliable. We present an alternative definition of the search direction that is not based on the minimizer of QP.

Our goals can be summarized as being:

- The derivation of a global convergence proof for the algorithm.
- A proof of superlinear convergence under additional assumptions.
- A proof that the penalty parameter used in the merit function is bounded.

Incomplete solutions for QP subproblems

The great majority of SQP algorithms in the literature define the search direction from a minimizer of the QP subproblem, although there have been some proposals to terminate the solution process for the QP subproblems early. An approach solving QP subproblems inexactly is described in Dembo and Tulowitzki [DT85], where for a generic SQP algorithm an early termination rule is given in terms of the norm of the reduced gradient for the subproblem. This rule gives a search direction p_k satisfying the condition

$$\|p_k - p_k^*\| = o(\|p_k\|),$$

where p_k^* denotes the minimizer for the kth QP subproblem.

We follow a different approach, presenting an early termination rule that is constructive in nature, and that has a guaranteed bound on the effort necessary to satisfy it. If the solution process is terminated early, the search direction for the outer iteration (the step in the original variables) is defined as the "total" step taken in the QP subproblem up to that point. The characteristics of the point at which the termination takes place clearly depend on the specific strategy used to solve the QP subproblem. In the course of solving a QP an active-set method generates iterates that are stationary points. We show that such points may be used to construct a suitable search direction.

Terminating the QP algorithm prior to obtaining a solution impacts the SQP algorithm in a number of critical ways. Not only is the search direction different, but also the QP multipliers will not be available. The merit function of interest requires the definition of a search direction in the space of the multipliers. In NPSQP this direction is defined using the QP multipliers. The proof of convergence for NPSQP makes use of the fact that the

QP multipliers are positive. The consequences of terminating the QP solution process early are therefore far reaching.

The remainder of this paper is organized as follows. Section 2 describes the form of the general algorithm, and presents the conditions on the search direction and the multiplier estimate. Section 3 studies the convergence properties of the algorithm; it is shown that such an algorithm is globally convergent. In Section 4 we show that the algorithm converges superlinearly. We also show that the penalty parameter used in the merit function is bounded. Section 5 establishes the convergence of algorithms that use merit functions belonging to a certain class (that includes the ℓ_1 merit function, for example). Finally, Section 6 presents numerical results obtained from an implementation of the first algorithm.

2. Description of the algorithm

This section introduces the proposed algorithm. The algorithm obtains the search direction from an incomplete solution of QP. The iterates are determined by conducting a linesearch on the following merit function:

$$\boxed{L_A(x,\lambda,s,\rho) = F(x) - \lambda^T(c(x)-s) + \tfrac{1}{2}\rho(c(x)-s)^T(c(x)-s)\,,} \tag{2.1}$$

where $s \geq 0$ are slack variables, and the scalar ρ is known as the penalty parameter.

The search is performed on an expanded space, including the Lagrange multiplier estimates λ, and the slack variables s. The symbols p, ξ and q will be used to denote the components of the search direction on the corresponding subspaces. In this case, the value of the merit function as a function of the steplength will be denoted by

$$\phi(\alpha) \equiv L_A(x+\alpha p, \lambda+\alpha\xi, s+\alpha q, \rho). \tag{2.2}$$

The derivative of ϕ with respect to α is denoted by ϕ'.

The following conventions will be used in the rest of the paper:

$$g_k \equiv \nabla F(x_k), \qquad A_k \equiv \nabla c(x_k), \qquad c_k \equiv c(x_k),$$

although the last two symbols, A_k and c_k, will also be used with the same meaning but restricted to the set of active constraints at the given point. The term *active constraint* will be used to designate a constraint that is satisfied exactly at the current point ($c_i(x) = 0$ in NP, or $a_i^T p = -c_i$ in QP), and the set of all constraints active at a given point will be referred to as the *active set* at the point.

The objective function for the QP subproblem will be denoted by $\psi_k(p)$,

$$\psi_k(p) \equiv g_k^T p + \tfrac{1}{2} p^T H_k p.$$

Sometimes, ψ will denote the function of one variable $\psi_k(\alpha) \equiv \psi_k(p+\alpha d)$. Finally, symbols of the form β_{abc} indicate fixed scalars related to properties of the problem, or the implementation of the algorithm, where "*abc*" identifies the specific scalar represented.

The algorithm

The following is an outline of the algorithm, the details are given in subsequent sections.

Let $\{H_k\}$ denote a sequence of symmetric matrices whose smallest eigenvalue is positive and uniformly bounded away from zero and whose largest eigenvalue is uniformly bounded.

The algorithm proceeds through the following steps:

(i) Set $\rho_{-1} = 0$. Choose x_0, λ_0 and $\rho_0 \ge 0$. Set $k = 0$.

(ii) At each point x_k, form the QP subproblem

$$\begin{array}{ll} \underset{p\in\Re^n}{\text{minimize}} & g_k^T p + \tfrac{1}{2} p^T H_k p \\ \text{subject to} & A_k p \ge -c_k, \end{array}$$

and from an incomplete solution determine p_k satisfying certain conditions. Compute a vector of multipliers μ_k satisfying a second set of conditions. The precise conditions that p_k and μ_k need to satisfy are given later in this section. If $p_k = 0$, set $\lambda_k = \mu_k$ and terminate. Otherwise, compute $\xi_k = \mu_k - \lambda_k$.

(iii) Compute s_k from

$$(s_k)_i = \begin{cases} \max(0, (c_k)_i) & \text{if } \rho_{k-1} = 0, \\ \max\Big(0, (c_k)_i - \dfrac{(\lambda_k)_i}{\rho_{k-1}}\Big) & \text{otherwise.} \end{cases}$$

Find ρ_k such that $\phi_k'(0, \rho_k) \le -\tfrac{1}{2} p_k^T H_k p_k$.

Compute q_k from

$$q_k = A_k p_k + c_k - s_k. \tag{2.3}$$

(iv) Compute the steplength α_k. The termination conditions for the linesearch are as follows:

If

$$\phi(1) - \phi(0) \le \sigma\phi'(0) \tag{2.4}$$

set $\alpha_k = 1$. Otherwise, find an $\alpha_k \in (0,1)$ such that

$$\phi(\alpha_k) - \phi(0) \le \sigma\alpha_k\phi'(0) \tag{2.5a}$$

$$\phi'(\alpha_k) \ge \eta\phi'(0), \tag{2.5b}$$

where $0 < \sigma \le \eta < \tfrac{1}{2}$.

(v) Update x_k and λ_k using

$$\begin{pmatrix} x_{k+1} \\ \lambda_{k+1} \\ \bar{s}_k \end{pmatrix} = \begin{pmatrix} x_k \\ \lambda_k \\ s_k \end{pmatrix} + \alpha_k \begin{pmatrix} p_k \\ \xi_k \\ q_k \end{pmatrix}.$$

Set $k \leftarrow k + 1$ and repeat steps (ii) to (v) until convergence is reached.

The definition of the search direction

In each iteration the search direction is constructed from the information obtained when solving the QP subproblem by an active-set method. Information available at the point the algorithm is terminated is used to construct p_k in the way indicated by the following steps. We have omitted the subscript k corresponding to the iteration number.

(i) An initial feasible point p^0 for the QP subproblem is obtained.

If the minimizer for the QP is used to determine the search direction, then, given the uniqueness of p_k, the choice of p^0 is irrelevant. When an incomplete solution of the QP subproblem is used to define the search direction, the choice of p^0 becomes critical. If we determine the search direction from a stationary point that is not a minimizer, the sequence of stationary points that we compute depends directly on the value of p^0. We wish to define the initial point in such a manner that *all* stationary points are satisfactory points at which to terminate the solution process. We require $\|p^0\|$ to be small whenever the points x_k are close to $\hat{x}$, a stationary point of NP.

Define the vectors $\bar{s}$ and r to have components

$$\bar{s}_i \equiv \max(0, c_i - \mu_i),$$
$$r_i \equiv \begin{cases} c_i - s_i & \text{if } |c_i - s_i| < |c_i - \bar{s}_i|, \\ c_i - \bar{s}_i & \text{otherwise;} \end{cases}$$

where μ denotes a multiplier estimate such that the following property holds:

$$\|x_k - \hat{x}\| \to 0 \Rightarrow \|c_k - \bar{s}_k\| \to 0.$$

From this definition, r has the following property:

$$\|c^-\| \le \|r\| \le \|c - s\|. \tag{2.6}$$

The following condition on p^0 ensure our objective:

- For some constant $\beta_{pcs} > 0$,

$$\|p^0\| \le \beta_{pcs}\|r\|. \tag{2.7}$$

(ii) A sequence of feasible descent steps are taken and the QP algorithm is terminated at a stationary point $\hat{p}$ for the QP subproblem.

(iii) If the stationary point is a second-order KKT point of the QP subproblem, the search direction is computed as $p = \hat{p}$.

(iv) If the stationary point is not a second-order KKT point, the multiplier vector at the stationary point must have some components that are negative. In this case, we need to compute a vector d and a scalar α with the following properties:

- d is feasible with respect to the active constraints, $Ad \geq 0$, and its norm is bounded above and below, that is, for some constants $\beta_{und} > \beta_{lnd} > 0$ it holds that $\beta_{und} \geq \|d\| \geq \beta_{lnd}$. It is assumed without loss of generality that $\beta_{lnd} \leq 1$, in order to simplify the arguments in the proofs.
- The rate of descent along d is sufficiently large. If $\psi(\zeta) \equiv \psi(\hat{p}+\zeta d)$, it is required that
$$\psi'(0) = (H\hat{p} + g)^T d \leq -\beta_{dsc} \max_i \mu_i^- \tag{2.8}$$
for some constant $\beta_{dsc} > 0$.
- The step α is taken as the step to the minimizer of $\psi(\zeta)$, if it is feasible. We have
$$\alpha = \min(\alpha_c, \alpha_m, \alpha_M),$$
where α_c is the step to the nearest inactive constraint, $\alpha_M > 0$ is a specified bound on the largest acceptable step, and
$$\alpha_m = -\frac{\psi'(0)}{d^T H d}. \tag{2.9}$$

It is a desirable property to avoid having search directions with very small norms, unless the corresponding point is close to a solution. The following definition is sufficient to ensure this property:

$$p = \begin{cases} \hat{p} + \alpha d & \text{if } \|\hat{p}\| < \beta_{slp}\|\hat{p} + \alpha d\|, \\ \hat{p} & \text{otherwise,} \end{cases} \tag{2.10}$$

for some constant $\beta_{slp} > 0$. It will be assumed that β_{slp} is chosen so that $\beta_{slp} \geq 1$.

The multiplier estimates

Step (ii) of the algorithm requires not only a search direction p_k, but also an estimate μ_k of the Lagrange multipliers at the current point. The QP algorithm may terminate at a stationary point, so a natural choice would be to use the multipliers of the stationary point as the estimate, but in general these may not be the best possible choice, as they may be negative, or the active set associated with the search direction may not in some cases be the same as the one for which the multiplier was obtained. The following conditions on μ_k are sufficient to ensure that the algorithm has the desired convergence properties.

MC1. The estimates are uniformly bounded in norm.

MC2. $\|\mu_k - \lambda^*\| = O(\|p_k\|)$, where λ^* denotes the multiplier vector associated with the solution point closest to x_k.

MC3. The complementarity condition $\mu_k^T(A_k p_k + c_k) = 0$ is satisfied at all iterations.

Condition **MC1** and the form in which the multiplier estimates are updated imply $\{\lambda_k\}$ are uniformly bounded. This result is given as Lemma 4.2 in [GMSW86b].

Lemma 2.1. *For all $k \geq 0$,*

$$\|\lambda_k\| \leq \max_{-1 \leq j \leq k-1} \|\mu_j\|,$$

and hence $\|\lambda_k\|$ is bounded for all k (by convention $\mu_{-1} \equiv \lambda_0$).

Second-order information

The condition imposed on $\{H_k\}$ is that the matrices in the sequence are positive definite and bounded, with bounded condition number. This assumption is identical to the one made for NPSQP. In practice, such a sequence may be generated (see [GMSW86a]) by updating a quasi-Newton approximation to the Hessian of the Lagrangian function or the Hessian of the augmented Lagrangian function in each iteration together with certain safeguards.

From this condition, some quantities will be uniformly bounded in the algorithm. The notation introduced below is used throughout for these bounds.

β_{lvH} is an upper bound for the largest eigenvalue of H: $p^T H p \leq \beta_{lvH} \|p\|^2$.

β_{svH} is a positive lower bound for the smallest eigenvalue of H: $p^T H p \geq \beta_{svH} \|p\|^2$.

3. Global convergence results

The results in this section establish global convergence properties for the SQP algorithm under study.

Assumptions and bounds

We make the following assumptions:

A1. x_k lies in a closed, bounded region $\Omega \subset \Re^n$, for all k.

A2. F, c_i and their first and second derivatives are continuous and uniformly bounded in norm on Ω.

A3. The Jacobian corresponding to the active constraints at any limit point of the sequence generated by the algorithm has full rank.

A4. A feasible point exists to all the QP subproblems.

A5. Strict complementarity holds at all stationary points of NP in Ω.

A6. The reduced Hessian of the Lagrangian function is nonsingular at all first-order KKT points of NP.

From these assumptions, several quantities are uniformly bounded in the algorithm. The first two bounds follow from assumption **A2**; the third follows from **A3**.

β_{nmc} is a bound for the norm of the constraint vector: $\|c_k\| \leq \beta_{nmc}$.

β_{nmg} is a bound for the norm of the gradient: $\|g_k\| \leq \beta_{nmg}$.

β_{nmu} is an upper bound for the norm of the multipliers corresponding to a minimizer of the QP subproblem: $\|\tilde{\mu}_k\| \leq \beta_{nmu}$.

Properties of the search direction

The first set of results explores the relationship of stationary points of the QP subproblems and stationary points of NP. The significance of this relationship is due to the fact that the search direction is obtained from information available at a stationary point of the QP subproblem. The results shown below are similar in spirit to those in Robinson [Rob74]. They will be used to show that the value of $\|p_k\|$ is "small" if and only if we are close to a solution point, with corresponding implications regarding the identification of the correct active set.

Lemma 3.1. *For any $x \in \Omega$, let p be a stationary point of the QP subproblem at x. For all $\epsilon > 0$ there exist a constant $\delta > 0$ and a point $\hat{x}$ such that*

$$\|p\| \leq \delta \Rightarrow \|x - \hat{x}\| \leq \epsilon,$$

where $\hat{x}$ is a stationary point for the NP, with the same set of active constraints as p, if the Jacobian of the active constraints at $\hat{x}$ is nonsingular.

Proof. Assume that the result does not hold; then there exist sequences $\{p_k\}_{k=1}^{\infty}$, and $\{x_k\}_{k=1}^{\infty}$, such that p_k is a stationary point of the QP subproblem at x_k satisfying $\|p_k\| \to 0$, and $\|x_k - \hat{x}\| > \epsilon$ for some $\epsilon > 0$.

A convergent subsequence can be extracted from $\{x_k\}$, using the compactness of Ω. Select now a sub-subsequence having fixed active set that is a subset of the active set at the limit point $\tilde{x}$. Such a subseqence must exist since there only a finite number of choices of active sets.

If we take limits along this subsequence of

$$A_k p_k + c_k \geq 0$$

it follows from $\|p_k\| \to 0$ and assumption **A2** that $\tilde{x}$ must be feasible.

If the Jacobian of the active constraints is nonsingular at $\hat{x}$, it follows from

$$H_k p_k + g_k = A_k^T \mu_k$$

that there exists a subsequence along which $\{\mu_k\}$ converges, $\mu_k \to \mu$. Taking limits along this subsequence,

$$\tilde{g} = \tilde{A}^T \tilde{\mu}.$$

This result implies that $\tilde{x}$ is a stationary point of NP, contradicting the assumption.

To show that the set of active constraints should be the same for p and $\hat{x}$, assume that sequences as described above exist, but that the set of active constraints at each p_k is not the same as the set of active constraints at $\hat{x}$. As $\|p_k\| \to 0$, the set of active constraints at each p_k must be a subset of the active constraints at $\hat{x}$; but if it is a proper subset, then there must exist an index i, active at $\hat{x}$, such that $(\mu_k)_i = 0$ for large enough k, and this will imply $\tilde{\mu}_i = 0$, violating the strict complementarity assumption. ∎

The assumptions on the form of the problem guarantee that large enough steps can be taken from stationary points in the QP subproblems when the points considered are not close to solutions for the problem. The algorithm makes use of this property to move away from stationary points for NP. The next result establishes the existence of some of the necessary bounds.

Lemma 3.2. *There exist positive values β_{spc} and β_{spm}, such that for all stationary points $\hat{x}$ that have inactive constraints*

$$\min_{i:\hat{c}_i>0} \hat{c}_i > \beta_{spc},$$

and for those stationary points having some negative multiplier element,

$$\max_i \hat{\mu}_i^- > \beta_{spm}.$$

Proof. Assume that there exists a sequence $\{\hat{x}_k\}$ of stationary points of NP in Ω such that

$$\min_{i:\hat{c}_{k_i}>0} \hat{c}_{k_i} \to 0.$$

From the compactness of Ω, a convergent subsequence can be extracted having fixed active set, and such that the minimum is always achieved for the same constraint (or set of constraints). Let $\hat{x}^*$ denote the limit point. It follows from Lemma 3.1 and assumption **A3** that $\hat{x}^*$ is a stationary point of NP. At this point assumption **A5** will be violated, as the corresponding constraints are active, but have zero multipliers.

If the sequence is such that

$$\max_i \hat{\mu}_{k_i}^- \to 0,$$

using the same construction assumption **A5** will again be violated at $\hat{x}^*$, as at least one of the multipliers corresponding to an active constraint will be zero. ∎

We now proceed to establish that if the norm of the search direction in any given iteration $\|p_k\|$ is small enough, then the correct active set must have been identified. If the norm of the stationary point where the search direction is computed, $\|\hat{p}_k\|$, is bounded away from zero, then (2.10) implies that $\|p_k\|$ is also bounded away from zero, and so in the proof of this result we consider only those iterations in which $\|\hat{p}_k\|$ is small.

From Lemma 3.1 and **A3** we know that if this norm is small, we must be close to, $\hat{x}$, a stationary point of NP, and in that case we can use the results from Lemma 3.2 to bound the size of the search direction.

Lemma 3.3. *There exists a value $\epsilon' > 0$ such that if $\|p_k\| \le \epsilon'$ then p_k is a minimizer of the QP subproblem and the correct active set at a solution has been identified. Also, $\|p_k\| = 0$ if and only if x_k is a first-order KKT point of NP.*

Proof. From Lemmas 3.1 and 3.2, there exist positive constants β_{spd} and δ^o such that if $\|\hat{p}_k\| \le \delta^o$ and $\hat{p}_k$ was not obtained as the minimizer of the QP subproblem, then

$$\psi(\hat{p}_k) - \psi(p_k) > \beta_{spd}, \tag{3.1}$$

and from the continuity of ψ, there exists a $\delta > 0$ such that $\|\hat{p}_k - p_k\| > \delta$.

Define

$$\beta_1^o \equiv \min(\delta^o, \frac{\delta}{2}).$$

If $\|\hat{p}_k\| \le \beta_1^o$, then

$$\|p_k\| \ge \|\hat{p}_k - p_k\| - \|\hat{p}_k\| \ge \frac{\delta}{2} \ge \beta_1^o.$$

If $\|\hat{p}_k\| > \beta_1^o$, then from (2.10),

$$\|p_k\| \ge \frac{\|\hat{p}_k\|}{\beta_{slp}} > \frac{\beta_1^o}{\beta_{slp}},$$

and thus in all cases the final point obtained has norm bounded away from zero.

If p_k is obtained from the minimizer of the QP subproblem, then Lemma 3.1 can be used directly. Assume that a sequence of points $\{x_k\}$ exists such that $\|p_k\| \to 0$, and all p_k are obtained as the solutions of the corresponding QP subproblems, but the active sets do not correspond to the one at a solution. By extracting a subsequence having a fixed active set and taking limits, a solution for the original problem with that active set is obtained (from assumption **A3**, it must hold that the multiplier vectors converge to the multipliers at the limit point), contradicting the hypothesis. Hence, a lower bound for $\|p_k\|$ must also exist in this case.

For the second part of the lemma it follows from the previous remarks that $p_k = 0$ if and only if p_k is a solution of the QP subproblem. Furthermore,

$$\begin{aligned} p_k \equiv 0 \text{ is a solution of QP} &\Leftrightarrow g_k = A_k^T\mu_k,\ \mu_k \ge 0,\ c_k \ge 0,\ \mu_k^T c_k = 0 \\ &\Leftrightarrow x_k \text{ is a first-order KKT point of NP,} \end{aligned}$$

completing the proof. ∎

Equivalence of sequences

For a given sequence $\{x_k\}$, the next result establishes the equivalence between the sequences $\{x_k - x^*\}$ and $\{p_k\}$.

Lemma 3.4. *If x^* denotes the solution point closest to x_k, then there exists a constant M_p, independent of k, such that*

$$\|x_k - x^*\| \le M_p\|p_k\|. \tag{3.2}$$

Proof. The proof is in essence the one given for Lemma 4.1 in [GMSW86b]. ∎

Descent properties

We need to impose some condition on the direction p_k to ensure that adequate descent can be obtained in each iteration. To be precise, it is necessary to satisfy in each iteration the bound on the directional derivative given in step (iii) of the algorithm.

The next lemma shows that if the starting point for the QP subproblem is selected as indicated in Section 2, the search direction satisfies a certain bound that will be shown to imply the desired result. Recall that r_k was the quantity introduced in Section 2 to provide a bound for the norm of the initial point p_k^0, and that its most relevant property for the proofs that follow is its relationship to $c_k - s_k$, given by (2.6).

Lemma 3.5. *There exist constants $\beta_1 > 0$, $\beta_2 \geq 0$, and initial points for the QP subproblem that give values for p_k, the search direction, satisfying*

$$\psi(p_k) = p_k^T g_k + \tfrac{1}{2} p_k^T H_k p_k \leq -\beta_1 \|p_k\|^2 + \beta_2 \|r_k\|. \tag{3.3}$$

Proof. In the proof we drop the subscript corresponding to the iteration number. Consider the following cases:

(i) p is the minimizer of the QP subproblem. Then,

$$p^T g + p^T H p = p^T A^T \bar{\mu} = -c^T \bar{\mu} \leq -\bar{\mu}^T c^- \leq \|\bar{\mu}\| \|c^-\|$$

$$\psi \leq -\tfrac{1}{2} p^T H p + \beta_{nmu} \|c^-\|,$$

where $\beta_{nmu} > 0$ is a bound on the norm of the QP multipliers. Note that from the properties of H, $p^T H p \geq \beta_{svH} \|p\|^2$.

(ii) p is obtained from a stationary point $\hat{p}$ of QP that is not a minimizer. Let δ^o be the value introduced in the proof of Lemma 3.3. One of the following three cases must apply:

- Assume that $\|\hat{p}\| > \delta^o$ and $\|\hat{p} - p^0\| \leq \frac{1}{2}\delta^o$. From these conditions, $\|p^0\| > \frac{1}{2}\delta^0$.

  From $\|p^0\| \leq \beta_{pcs} \|r\|$, assumptions **A1–A2** and Lemma 2.1, it follows that $\|p^0\| \leq K$ and

  $$\psi(p^0) \leq \beta_{nmg} K + \tfrac{1}{2} \beta_{lvH} K^2 = \bar{K}.$$

  We then have $\psi(p) \leq \bar{K}$, implying

  $$\tfrac{1}{2}(p + H^{-1}g)^T H (p + H^{-1}g) - \tfrac{1}{2} g^T H^{-1} g \leq \bar{K},$$

  and hence

  $$\|p + H^{-1}g\|^2 \leq \frac{2\bar{K}\beta_{svH} + \beta_{nmg}^2}{\beta_{svH}^2},$$

giving the bound

$$\|p\| \le \beta_{nmp} = \frac{\beta_{nmg}}{\beta_{svH}} + \sqrt{\frac{2\bar{K}\beta_{svH} + \beta_{nmg}^2}{\beta_{svH}^2}}.$$

Using this property, we can write

$$\|p\| \le \beta_{nmp} \qquad \frac{2\beta_{nmp}}{\delta^0}\|p^0\|.$$

Defining $\beta_2^o = \beta_{nmg} + \beta_{lvH}\beta_{nmp}$, we have

$$p^T g + p^T H p \le \beta_2^o\|p\| \le \frac{2\beta_2^o\beta_{nmp}}{\delta^0}\|p^0\| \le \frac{2\beta_2^o\beta_{nmp}\beta_{pcs}}{\delta^0}\|r\|,$$

giving finally

$$\psi \le -\tfrac{1}{2}p^T H p + \frac{2\beta_2^o\beta_{nmp}\beta_{pcs}}{\delta^o}\|r\|.$$

- Assume that $\|\hat{p} - p^0\| > \frac{1}{2}\delta^o$. Let p_i denote the ith iterate for the QP subproblem and $\psi_i = \psi(p_i)$. We have

$$\psi_{i-1} - \psi_i = -\alpha_i(g^T d_i + p_{i-1}^T H d_i) - \tfrac{1}{2}\alpha_i^2 d_i^T H d_i = d_i^T H d_i\,\alpha_i(1 - \tfrac{1}{2}\alpha_i).$$

Summing over all the iterations to the stationary point, and letting $\hat{\psi} = \psi(\hat{p})$ we get

$$\psi_0 - \hat{\psi} = \textstyle\sum_i d_i^T H d_i\,\alpha_i(1 - \frac{1}{2}\alpha_i) \ge \beta_{svH}\sum_i\|d_i\|^2\alpha_i(1 - \frac{1}{2}\alpha_i),$$

but from $\|\hat{p} - p^0\| = \|\sum_i \alpha_i d_i\| > \frac{1}{2}\delta^o$, for at least one i we must have

$$\alpha_i\|d_i\| > \frac{\delta^o}{2m},$$

where m is a bound on the number of steps; using $\alpha_i \le 1$, it must hold that

$$\psi_0 - \hat{\psi} \ge \beta_{svH}\left(\frac{\delta^o}{2m}\right)^2\left(\frac{1}{\alpha_i} - \frac{1}{2}\right) \ge \bar{\gamma} = \tfrac{1}{2}\beta_{svH}\left(\frac{\delta^o}{2m}\right)^2. \tag{3.4}$$

From

$$\psi_0 = g^T p^0 + \tfrac{1}{2}p^{0T} H p^0 \le \beta_2^o\|p^0\| \le \beta_{pcs}\beta_2^o\|r\| \tag{3.5}$$

we can derive the following bound:

$$\psi \le \hat{\psi} \le \psi_0 - \bar{\gamma} \le -\beta_1\|p\|^2 + \beta_{pcs}\beta_2^o\|r\|$$

for $0 < \beta_1 \le \bar{\gamma}/\beta_{nmp}^2$.

- If $\|\hat{p}\| \le \delta^o$, then from (3.1),

$$\psi_0 - \psi > \beta_{spd},$$

and using (3.5)

$$\psi \le -\beta_{spd} + \beta_{pcs}\beta_2^o\|r\| \le -\beta_1\|p\|^2 + \beta_{pcs}\beta_2^o\|r\|,$$

where $0 < \beta_1 \le \beta_{spd}/\beta_{nmp}^2$. ∎

Definition of the penalty parameter

We now show that the penalty parameter can be selected in such a way that the initial descent available for the linesearch is sufficiently large. We start by defining a scalar $\hat{\rho}$. It will be shown in a subsequent lemma that $\phi_k'(0,\hat{\rho})$ satisfies the bound given in step (iii) of the algorithm.

We first introduce a vector b and a nonnegative scalar β_2'. Their values should be selected as follows:

- Define $\bar{\mu}_k$ as the QP multipliers if p_k is the minimizer of the QP subproblem; otherwise define $\bar{\mu}_k$ as a multiplier estimate satisfying conditions **MC1–MC3**.
- Define
$$b \equiv \begin{cases} \mu & \text{if } p^Tg + \mu^T(c-s) \le -p^THp, \\ \bar{\mu} & \text{otherwise.} \end{cases}$$
- Define
$$\beta_2' \equiv \max(0, \hat{\beta}_2),$$
where
$$\hat{\beta}_2\|c-s\|_1 = p^Tg + \tfrac{1}{2}p^THp + b^T(c-s).$$

Note that β_2' is bounded since from Lemma 3.5 we have,

$$p^Tg + \tfrac{1}{2}p^THp + b^T(c-s) \le p^Tg + p^THp + b^T(c-s) \le (\beta_2 + \|b\|)\|c-s\|.$$

The value for β_2' presented above is related to the constant introduced in (3.3), while the value of b is related to the QP multipliers at the current point. For the purpose of satisfying the bound given in step (iii) of the algorithm, b can be taken to be zero, but as it will be seen later, it plays an important role in ensuring that the penalty parameter is chosen in a way that does not inhibit superlinear convergence.

Define
$$\hat{\rho} \equiv \frac{\|2\lambda - \mu - b + \beta_2' v\|}{\|c-s\|}, \tag{3.6}$$
where $v_i \equiv \text{sign}(c_i - s_i)$, giving $v^T(c-s) = \|c-s\|_1$.

Lemma 3.6. *For all $\rho \ge \hat{\rho}_k$, where $\hat{\rho}_k$ is the value defined in (3.6),*
$$\phi_k'(0,\rho) \le -\tfrac{1}{2}p_k^TH_kp_k. \tag{3.7}$$

Proof. Again, we drop the subscript corresponding to the iteration number.

We start by introducing an expression for $\phi'(0)$. To derive it, consider first the gradient of L_A with respect to x, λ and s,

$$\nabla L_A(x,\lambda,s) \equiv \begin{pmatrix} g(x) - A(x)^T\lambda + \rho A(x)^T(c(x)-s) \\ -(c(x)-s) \\ \lambda - \rho(c(x)-s) \end{pmatrix}. \tag{3.8}$$

It follows that $\phi'(0)$ is given by

$$\begin{aligned}\phi'(0) &= p^Tg - p^TA^T\lambda + \rho\, p^TA^T(c-s) - (c-s)^T\xi + \lambda^Tq - \rho q^T(c-s)\\ &= p^Tg + (2\lambda-\mu)^T(c-s) - \rho\|c-s\|^2,\end{aligned} \tag{3.9}$$

where g, A, and c are evaluated at x.

From (3.9) and (3.6),

$$\begin{aligned}\phi'(0,\hat\rho) &= p^Tg + (2\lambda-\mu)^T(c-s) - \|c-s\|\|2\lambda-\mu-b+\beta_2'v\|\\ &\le p^Tg + b^T(c-s) - \beta_2'v^T(c-s)\\ &\le -\tfrac{1}{2}p^THp,\end{aligned}$$

completing the result. ∎

An immediate consequence of (3.7) and the properties of H_k is the following bound on the directional derivative:

$$\phi_k'(0) \le -\tfrac{1}{2}\beta_H\|p_k\|^2, \tag{3.10}$$

for $\beta_H \le \beta_{svH}$.

The strategy for the selection of the penalty parameter ρ_k is to define its value to satisfy (3.7), while remaining small enough to be bounded by a multiple of $\hat\rho$. An example of a selection rule having these properties is as follows.

Let

$$\rho_k = \begin{cases}\rho_{k-1} & \text{if } \phi'(0,\rho_{k-1}) \le -\frac{1}{2}p_k^TH_kp_k,\\ \max(\hat\rho_k, 2\rho_{k-1}) & \text{otherwise,}\end{cases} \tag{3.11}$$

where $\hat\rho_k$ is defined as in (3.6). Then, for any iteration k_l in which the parameter needs to be increased, it holds that $\rho_{k_l} \ge 2\rho_{k_{l-1}}$. It follows from this result and (3.6) that the penalty parameter goes to infinity if and only if its value is increased in an infinite number of iterations.

For the rest of this paper, the symbol ρ_k will denote a value of the penalty parameter satisfying $\hat\rho_k \le \rho_k \le K\hat\rho_k$, for some constant $K > 1$.

Proof of global convergence

The proof of global convergence is based on the fact that the decrease in the value of the merit function in each iteration is bounded away from zero by a sufficiently large value, related to the norm of the search direction. This fact is implied by the existence of certain bounds on the rate of growth for the penalty parameter, introduced in the following lemma. The notation k_l is used in all that follows to indicate iterations at which the value of the penalty parameter needs to be modified.

Lemma 3.7. *For any iteration k_l in which the value of ρ is modified,*

$$\rho_{k_l}\|c_{k_l} - s_{k_l}\| \le N$$

and

$$\rho_{k_l}\|p_{k_l}\|^2 \le N,$$

for some constant N.

Proof. All quantities in the proof refer to iteration k_l, and so this subscript is dropped.

From the boundedness of β_2', Lemma 2.1, the definition of $\bar{b}$, and condition **MC1** on the multipliers, there must exist a constant N_1 such that

$$\|2\lambda - \mu - \bar{b} + \beta_2' v\| \le N_1,$$

and from the definition of $\hat{\rho}$ and the condition that ρ has to be selected as a finite multiple of $\hat{\rho}$,

$$\rho\|c - s\| \le N_1.$$

For the second part, using Lemma 3.5 (we add the term $\bar{b}^T(c-s)$ using the boundedness of $\|\bar{b}\|$), we can write after some algebraic manipulation

$$\begin{aligned}\phi'(0) &= p^T g + (2\lambda - \mu)^T(c - s) - \rho\|c - s\|^2 \\ &\le -\tfrac{1}{2}p^T H p - \beta_1\|p\|^2 + (2\lambda - \mu - \bar{b} + \beta_2 v)^T(c - s) - \rho(c - s)^T(c - s).\end{aligned}$$

Since we have $\phi'(0, \rho^-) > -\frac{1}{2}p^T H p$, where ρ^- is the value of ρ at the previous iteration, it follows that

$$\begin{aligned}\beta_1\|p\|^2 &\le (2\lambda - \mu - \bar{b} + \beta_2 v)^T(c - s) \\ &\le \|2\lambda - \mu - \bar{b} + \beta_2 v\|\,\|c - s\|.\end{aligned}$$

We reorder terms to obtain

$$\|c - s\| \ge \beta_1 \frac{\|p\|^2}{\|2\lambda - \mu - \bar{b} + \beta_2 v\|}. \tag{3.12}$$

Multiplying both sides by ρ and using the same arguments as in the first part of the lemma yields

$$\rho\|p\|^2 \le N_2,$$

completing the proof. ∎

The convergence proof follows along lines similar to those presented in [GMSW86b]. The following theorem relies on results presented in this reference, that hold with only minor modifications for the algorithm considered in this paper.

Theorem 3.1. *The algorithm has the property that*

$$\lim_{k\to\infty} \|x_k - x^*\| = 0.$$

Proof. It follows from Lemma 3.4 that it is sufficient for the proof of the theorem to show

$$\lim_{k\to\infty} \|p_k\| = 0. \tag{3.13}$$

If $\|p_k\| = 0$ for any k, the algorithm terminates and the theorem is true. Hence we assume that $\|p_k\| \neq 0$ for any k.

Consider first the case when there is no upper bound on the penalty parameter. The following result, given as Lemma 4.6 in [GMSW86b], and as Lemma 3.6.3 in [Pr89], holds for the algorithm:

$$\rho_{k_l} \sum_{k=k_l}^{k_{l+1}-1} \|\alpha_k p_k\|^2 < M, \tag{3.14}$$

where M is a positive constant. It also holds that there exists a uniform lower bound on α_k, $0 < \bar{\alpha} \le \alpha_k$, (Lemma 4.9 in [GMSW86b], and Lemma 3.6.6 in [Pr89]). From these properties, (3.14) implies that for any $\delta > 0$ we can find an iteration index K such that

$$\|p_k\| \le \delta \quad \text{for} \quad k \ge K,$$

which implies that $\|p_k\| \to 0$ as required.

In the bounded case, we know that there exists a value $\tilde{\rho}$ and an iteration index $\tilde{K}$ such that $\rho = \tilde{\rho}$ for all $k \ge \tilde{K}$. We consider henceforth only such values of k.

The proof is by contradiction. We assume that there exists $\epsilon > 0$ and an infinite subsequence $\{k_i\}$ such that $\|p_{k_i}\| \ge \epsilon$ for all i. Consider only indices i such that $k_i > \tilde{K}$. Every iteration after $\tilde{K}$ must yield a strict decrease in the merit function because the boundedness of the steplength implies

$$\phi(\alpha) - \phi(0) \le \sigma\alpha\phi'(0) \le -\tfrac{1}{2}\sigma\bar{\alpha}\beta_H\|p\|^2 < 0.$$

The adjustment of the slack variables s in step (ii) of the algorithm can only lead to a further reduction in the merit function, as L is quadratic in s and the minimizer with respect to s_i is given by $c_i - \lambda_i/\rho$. From the fact that the penalty parameter is not modified, for iterations from the subsequence we have

$$\phi(x_{k_{i+1}}) - \phi(x_k) < \phi(x_{k_i+1}) - \phi(x_k) \le -\tfrac{1}{2}\sigma\bar{\alpha}\beta_H\epsilon^2.$$

Therefore, since the merit function with $\rho = \tilde{\rho}$ decreases by at least a fixed quantity at every step in the subsequence, it must be unbounded below. But this is impossible, from assumptions **A1**, **A2** and Lemma 2.1, so (3.13) must hold. ∎

Once the global convergence of the algorithm has been established, the next step is to show that the multiplier estimate λ_k also converges to the desired value. The result presented below, given as Theorem 4.2 in [GMSW86b], implies that the convergence of the multiplier estimates is a consequence of the global convergence of the algorithm, and the facts that the multiplier estimates are bounded in norm, and the steplength is bounded away from zero.

Corollary 3.1. *Let λ^* denote the multiplier vector at x^*. Then*

$$\lim_{k\to\infty} \|\lambda_k - \lambda^*\| = 0.$$

4. Rate of convergence

Under additional assumptions we shall show that the algorithm converges at a superlinear rate. In order to do this we first prove a number of results on the rate of growth of the penalty parameter ρ.

Lemma 4.1. *If there exists an infinite subsequence of iterations $\{k_l\}$ at which the penalty parameter is increased, then*

$$\lim_{l\to\infty} \rho_{k_l}\|c_{k_l} - s_{k_l}\| = 0$$

and

$$\lim_{l\to\infty} \rho_{k_l}\|p_{k_l}\|^2 = 0.$$

Proof. We drop the subscript k_l in what follows. From definition (3.6) and boundedness of the ratio $\rho/\hat{\rho}$,

$$\rho\|c - s\| \le 2\|2\lambda - \mu - \bar{b} + \beta_2' v\|,$$

and from the definition of $\bar{b}$ after Lemma 3.6,

$$\bar{b}_{k_l} \to \lambda^*.$$

As the QP multipliers satisfy $p^T g + p^T H p = -c^T\tilde{\mu}$, and for ρ large enough p is obtained as the solution of the QP subproblem, $\bar{b}$ eventually satisfies

$$p^T g + \bar{b}^T(c - s) \le -p^T H p,$$

implying that we can take $\beta_2' = 0$ in (3.6).

We can now use Corollary 3.1 to conclude that

$$\lim_{l\to\infty} \|2\lambda_{k_l} - \mu_{k_l} - \bar{b}_{k_l} + \beta_{2_{k_l}}' v_{k_l}\| = 0$$

and

$$\lim_{l\to\infty} \rho_{k_l}\|c_{k_l} - s_{k_l}\| = 0. \tag{4.1}$$

Finally, from (3.12) and (4.1) we have

$$\lim_{l\to\infty} \rho_{k_l}\|p_{k_l}\|^2 = 0,$$

completing the proof. ∎

Lemma 4.2. *If there exists an infinite subsequence $\{k_l\}$, then*

$$\lim_{l\to\infty} \rho_{k_l}\Big(\phi_{k_l}(\rho_{k_l}) - \phi_{k_{l+1}}(\rho_{k_l})\Big) = 0.$$

Proof. To simplify the notation we shall use the subscripts 0 and K to denote quantities associated with the iterations k_l and k_{l+1} respectively. Thus, the penalty parameter is increased at x_0 and x_K and remains fixed at ρ_0 for iterations $1, \dots, K-1$.

From the boundedness of $\|\lambda\|$ (Lemma 2.1), and the fact that $\rho_0 < \rho_K$, we have

$$\rho_0|\lambda_0^T(c_0 - s_0)| \leq 2\|\lambda_0\|\, \rho_0\|c_0 - s_0\| \to 0,$$
$$\rho_0|\lambda_K^T(c_K - s_K)| \leq 2\|\lambda_K\|\, \rho_K\|c_K - s_K\| \to 0,$$

and from Lemma 4.1 we have

$$\rho_0(\phi_0 - \phi_K) - \rho_0(F_0 - F_K) \to 0. \tag{4.2}$$

Using

$$\rho_0 c_0^T\tilde{\mu}_0 < \rho_0 K\|p_0\|^2 + \rho_0(c_0 - s_0)^T(2\lambda_0 - \mu_0). \tag{4.3}$$

we have

$$\rho_0 K\|p_0\|^2 + \rho_0(c_0 - s_0)^T(2\lambda_0 - \tilde{\mu}_0) > \rho_0 c_0^T\tilde{\mu}_0 \geq \rho_0(c_0 - s_0)^T\tilde{\mu}_0. \tag{4.4}$$

Using again Lemma 4.1, from (4.4) and assumption **A3**, implying the boundedness of $\|\tilde{\mu}_0\|$, we get

$$\rho_0 c_0^T\tilde{\mu}_0 \to 0. \tag{4.5}$$

From (2.6) (keeping the same notation),

$$-\rho_0 c_K^T\tilde{\mu}_0 \leq \rho_0 c_K^{-T}\tilde{\mu}_0 \leq \rho_0\|\tilde{\mu}_0\|\|c_K - s_K\| \to 0. \tag{4.6}$$

We can again use Lemma 4.1 to obtain

$$\rho_0 O\Big(\max(\|p_0\|^2, \|p_K\|^2)\Big) \to 0. \tag{4.7}$$

From (4.2), (4.5), (4.6) and (4.7) we obtain

$$\rho_0(\phi_0 - \phi_K) \to 0,$$

giving the desired result. ∎

Lemma 4.3. *For general iterations k,*

$$\lim_{k\to\infty} \rho_k\|p_k\|^2 = 0.$$

Proof. We use the same notation as lemma 4.2.

If ρ is bounded, the result follows from Theorem 3.1. If ρ is increased in an infinite subsequence of iterations, then for $0 \leq k \leq K-1$, property (2.5a) imposed by the choice of α_k, and the fact that the penalty parameter is not increased, imply that

$$\phi_k - \phi_{k+1} \geq -\sigma\alpha_k\phi_k'. \tag{4.8}$$

We can then write

$$\tfrac{1}{2}\sigma\beta_H \sum_{k=0}^{K-1} \alpha_k \|p_k\|^2 \le \phi_0 - \phi_K.$$

Rearranging this expression and using the property that $0 < \bar{\alpha} < \alpha_k$, we obtain

$$\tfrac{1}{2}\sigma\beta_H\bar{\alpha} \sum_{k=0}^{K-1} \|p_k\|^2 \le \phi_0 - \phi_K, \tag{4.9}$$

and the result follows from Lemma 4.2. ∎

Lemma 4.4. *For general iterations k,*

$$\lim_{k\to\infty} \rho_k \|c_k - s_k\| = 0.$$

Proof. If ρ is bounded the result follows from $c^* \ge 0$, $\lambda^* \ge 0$, $\lambda^{*T}c^* = 0$, Corollary ??, Lemma 3.1 and

$$c_i - s_i = \min(c_i, \frac{\lambda_i}{\rho}).$$

If ρ is increased in an infinite subsequence of iterations, consider two cases:

(i) If i is such that $c_i^* > 0$, then $\lambda_i^* = 0$ and as

$$\rho|c_i - s_i| = |\min(\rho c_i, \lambda_i)|,$$

from the convergence of the multiplier estimates, eventually $\rho|c_i - s_i| = |\lambda_i| \to 0$.

(ii) For those i such that $c_i^* = 0$, implying $\lambda_i^* > 0$, consider iteration indices large enough so that the correct active set is identified, implying $a_i^T p + c_i = 0$. Then, from the Taylor series expansion for c and the boundedness of the steplength,

$$\bar{c}_i = c_i + \alpha_0 a_i^T p + O(\|\alpha_0 p_0\|^2) = (1 - \alpha_0)c_i + O(\|p_0\|^2).$$

Recurring this relationship for the kth step between $k = 0$ and $k = K$ we get

$$\rho_k(c_k)_i = \rho_0(c_k)_i = \rho_0 \prod_{j=0}^{k-1}(1 - \alpha_j)(c_0)_i + \rho_0 O\Big(\sum_{j=0}^{k-1} \|p_j\|^2\Big),$$

but as $0 < \alpha_j \le 1$ we obtain

$$\rho_k|(c_k)_i| \le \rho_0|(c_0)_i| + \rho_0 O\Big(\sum_{j=0}^{k-1} \|p_j\|^2\Big). \tag{4.10}$$

From Lemma 4.1 we must have that $\rho_0|(c_0)_i| \to 0$, and using (4.10) and Lemma 4.3,

$$\rho_k|(c_k)_i| \to 0.$$

This completes the proof. ∎

Lemma 4.5. *For large enough k,*

$$\mu_k^T s_k = 0.$$

Proof. Assume k large enough so that the correct active set has been identified.

(i) If i is such that $c_i^* > 0$, from condition **MC3** on the multipliers, $\mu_{k_i} = 0$.

(ii) If i is such that $c_i^* = 0$, then, from strict complementarity, $\lambda_i^* > 0$. Also, from Lemma 4.4, $\rho_k((c_k)_i - (s_k)_i) = \min(\rho_k(c_k)_i, (\lambda_k)_i) \to 0$, so for large enough k, Lemma 3.1 will imply $\rho_k(c_k)_i \le (\lambda_k)_i$, and

$$(s_k)_i = \max\Big(0, (c_k)_i - \frac{(\lambda_k)_i}{\rho_k}\Big) = 0,$$

proving the result. ∎

To prove that the algorithm converges superlinearly it is necessary to assume that H_k converges to an approximation of $\nabla^2_{xx} L(x^*, \lambda^*)$ in some sense. We shall also introduce a stronger condition on the Lagrange multiplier estimate.

Define W_k as $W_k \equiv \nabla^2_{xx} L(x_k, \lambda_k)$. We impose the following condition on H_k:

A7. Following Boggs, Tolle and Wang [BTW82], we assume

$$\|Z_k^T (H_k - W_k) p_k\| = o(\|p_k\|),$$

where Z_k is a basis for the null space of A_k that is bounded in norm and has its smallest singular value bounded away from 0.

The proof proceeds by first showing that the sequence $\{x_k + p_k - x^*\}$ converges superlinearly, and then proving that a steplength of one is eventually attained. The results presented on bounds for the growth rate of the penalty parameter allow us to obtain an asymptotic expansion for the quantities involved in the linesearch termination criterion.

We want to prove that condition (2.4) is satisfied for all k large enough. We show in the following lemma that the satisfaction of (2.4) is directly related to the asymptotic properties of $T_k \equiv p_k^T (g_k - A_k^T \mu_k) + p_k^T W_k p_k$. In what follows, the absence of an argument indicates values at x_k, and an argument of 1 will indicate values at $x_k + p_k$.

Lemma 4.6. *The following relationship holds:*

$$\phi_k(1) - \phi_k = \tfrac{1}{2}\phi_k' + \tfrac{1}{2}T_k + o(\|p_k\|^2).$$

Proof. From (2.1) we have

$$\phi(1) - \phi = F(1) - F - \mu^T(c(1) - s - q) + \lambda^T(c - s) + \frac{\rho}{2}(c(1) - s - q)^T(c(1) - s - q) - \frac{\rho}{2}(c - s)^T(c - s).$$

Using the corresponding Taylor expansions around x_k,

$$c_i(1) - s_i - q_i = \tfrac{1}{2}p^T\nabla^2 c_i p + o(\|p\|^2),$$

we obtain

$$\phi(1) - \phi = g^T p + \tfrac{1}{2}p^T\nabla^2 F p - \tfrac{1}{2}\textstyle\sum_i \lambda_i p^T\nabla^2 c_i p - \tfrac{1}{2}\sum_i \xi_i p^T\nabla^2 c_i p + \lambda^T(c - s) + \frac{\rho}{8}\sum_i (p^T\nabla^2 c_i p)^2 - \frac{\rho}{2}(c - s)^T(c - s) + o(\|p\|^2). \tag{4.11}$$

From Lemma 4.3 and (4.11),

$$\begin{aligned}\phi(1) - \phi &= \phi' + \tfrac{1}{2}\left(p^T W p + 2\xi^T(c - s) + \rho(c - s)^T(c - s)\right) + o(\|p\|^2)\\ &= \tfrac{1}{2}\phi' + \tfrac{1}{2}\left(p^T W p + p^T g + \mu^T(c - s)\right) + o(\|p\|^2)\\ &= \tfrac{1}{2}\phi' + \tfrac{1}{2}\left(p^T W p + p^T(g - A^T\mu)\right) + o(\|p\|^2),\end{aligned}$$

completing the result. ∎

The assumptions made imply the superlinear convergence of the sequence $\{x_k + p_k - x^*\}$. The following lemma corresponds to Theorem 3.1 in [BTW82].

Lemma 4.7. *Under assumptions* **A1**-**A7**, *and conditions* **MC1**-**MC3**,

$$\|x_k + p_k - x^*\| = o(\|x_k - x^*\|). \tag{4.12}$$

The main result of this section is given in the next theorem, where it is shown that, if condition **MC2** is replaced by a stronger condition, then after a finite number of iterations a steplength of one is taken for all iterations thereafter, implying that the algorithm achieves superlinear convergence.

The new condition is

MC2'. $\|\mu_k - \lambda^*\| = o(\|x_k - x^*\|)$.

It is possible to prove Theorem 4.1 without the need to strengthen the conditions on the multipliers. It is shown in [Pr89] that there exists a constant M such that if $\rho_k > M$, condition **MC2** is sufficient.

Theorem 4.1. *Under assumptions* **A1**-**A7**, *and conditions* **MC1**, **MC2'** *and* **MC3**, *the algorithm converges superlinearly.*

Proof. As in Powell and Yuan [PY86], observe that the continuity of second derivatives gives the following relationships:

$$\begin{aligned} F(x+p) &= F(x) + \tfrac{1}{2}\Big(g(x) + g(x+p)\Big)^T p + o(\|p\|^2) \\ c(x+p) &= c(x) + \tfrac{1}{2}\Big(A(x) + A(x+p)\Big)p + o(\|p\|^2). \end{aligned}$$

From the Taylor series expansions we have

$$\begin{aligned} F(x+p) &= F(x) + g(x)^T p + \tfrac{1}{2} p^T \nabla^2 F(x) p + o(\|p\|^2) \\ c_i(x+p) &= c_i(x) + a_i(x)^T p + \tfrac{1}{2} p^T \nabla^2 c_i(x) p + o(\|p\|^2), \end{aligned}$$

and since (4.12) implies $g(x+p) = g^* + o(\|p\|)$, $a_i(x+p) = a_i^* + o(\|p\|)$, we get

$$\begin{aligned} p^T \nabla^2 F p &= (g^* - g)^T p + o(\|p\|^2) \\ p^T \nabla^2 c_i p &= (a_i^* - a_i)^T p + o(\|p\|^2). \end{aligned}$$

Given that $\sum_i \lambda_i p^T \nabla^2 c_i p = \sum_i \mu_i p^T \nabla^2 c_i p + o(\|p\|^2)$, we must have

$$p^T W p = p^T(g^* - A^{*T}\mu) - p^T(g - A^T\mu) + o(\|p\|^2). \tag{4.13}$$

Condition **MC2'** implies $p^T(g^* - A^{*T}\mu) = o(\|p\|^2)$, and from (4.13),

$$p^T W p + p^T(g - A^T\mu) = o(\|p\|^2). \tag{4.14}$$

From Lemma 4.6 and (4.14),

$$\phi(1) - \phi(0) = \tfrac{1}{2}\phi'(0) + o(\|p\|^2),$$

but from (3.10) condition (2.4) is eventually satisfied, and we have $x_{k+1} = x_k + p_k$ for all k large enough. In this case, from (4.12),

$$\lim_{k\to\infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} = 0,$$

i.e. superlinear convergence, completing the proof. ∎

Boundedness of the penalty parameter

The last result in this section shows that, if condition **MC2'** is replaced by a slightly stronger condition, the penalty parameter needs to be modified in at most a finite number of iterations (and consequently it remains bounded). The criterion presented will be satisfied, for example, by the least-squares multipliers computed at $x_k + p_k$.

Theorem 4.2. *If the multiplier estimates μ_k in the algorithm satisfy*

$$\|\mu_k - \lambda^*\| = O(\|x_k + p_k - x^*\|), \tag{4.15}$$

the penalty parameter is only increased a finite number of times, and there exists a constant M such that $\rho_k \le M$ for all k.

Proof. Let K be an iteration index such that, for all iterations $k \geq K$, p_k is determined as the minimizer of QP and $\alpha_k = 1$. We now consider iterations having $k > K$.

$$g_k^T p_k + p_k^T H_k p_k = p_k^T A_k \pi_k = -c_k^T \pi_k \leq -(c_k - s_k)^T \pi_k, \tag{4.16}$$

where π_k are the QP multipliers at iteration k. From (3.9), (4.16) and the fact that a unit steplength is accepted, it follows that

$$\begin{aligned} \phi_k'(0) &= g_k^T p_k + (2\lambda_k - \mu_k)^T (c_k - s_k) - \rho_k \|c_k - s_k\|^2 \\ &\leq \; - p_k^T H_k p_k + \|2\mu_{k-1} - \mu_k - \pi_k\| \|c_k - s_k\| - \rho_k \|c_k - s_k\|^2. \end{aligned}$$

If (4.15) holds, from the properties of H_k and $\|\pi_k - \lambda^*\| = O(\|p_k\|)$ we must have

$$\|2\mu_{k-1} - \mu_k - \pi_k\| \leq M p_k^T H_k p_k$$

for some positive constant M.

We can then write

$$\|2\mu_{k-1} - \mu_k - \pi_k\| \|c_k - s_k\| \leq \tfrac{1}{2} p_k^T H_k p_k + \tfrac{1}{2} M \|c_k - s_k\|^2,$$

implying that

$$\phi_k'(0) \leq -\tfrac{1}{2} p_k^T H_k p_k + (\tfrac{1}{2} M - \rho_k) \|c_k - s_k\|^2.$$

From this inequality it follows that if $\rho_k \geq \frac{1}{2} M$, condition (3.7) will be satisfied, and the penalty parameter will not need to be increased. Given that we are using the rule for updating ρ_k described after Lemma 3.6, it must hold that $\rho_k \leq \frac{1}{2} K M$. ∎

Note that the lemma does not require any conditions on how well H_k approximates W_k.

5. Other Merit Functions

Several merit functions have been proposed and analyzed in the literature (a review can be found in Powell [Po87]). The question arises if the convergence results using early termination in the solution of the QP subproblem depend on our specific merit function, or if they are fairly independent of this choice. We shall show in this section the choice of merit function is not critical. What we present is how to adapt our SQP algorithm to the use of other merit functions rather than examine other methods explicitly to see if the particular QP subproblem posed and the manner the search is performed can be adapted to the use of an incomplete solution. For example, we still perform a search in the x and λ spaces. Slack variables do not appear in the merit functions we shall consider, consequently the search in the space of the slack variables is no longer required. Because the freedom to choose the multiplier estimates is relaxed, our algorithm can be made to mimic other algorithms. For example, we are free to choose all the multiplier estimates to be zero making the search in the multiplier space specious.

We have selected as examples the study of two particular merit function. The first one corresponds to a class of merit functions that includes among others the ℓ_1 merit function analyzed in Han [Han76], Byrd and Nocedal [BN88] and Burke and Han [BH89]. This general merit function takes the form:

$$\phi(x,\lambda) = F(x) + \lambda^T c^-(x) + \rho\|c^-(x)\|, \tag{5.1}$$

where an ℓ_p norm ($1 \le p \le \infty$) is used, and $c_i^-(x) \equiv \max(0, -c_i(x))$. The second merit function we consider is

$$\phi(x,\lambda) = F(x) + \lambda^T c^-(x) + \tfrac{1}{2}\rho\|c^-(x)\|_2^2. \tag{5.2}$$

This merit function has been studied among others by Powell and Yuan [PY86]. Unlike the algorithm defined in [PY86], we do not explicitly define the form of the multiplier estimates although the one used in [PY86] satisfies the criteria on our estimates.

We still assume **A1–A6** hold for the problem. However, the multiplier estimate used, μ_k, is only required to satisfy **MC1** when the merit function (5.1) is used. This condition is trivial to satisfy. For example, we may choose $\lambda_0 = 0$ and $\xi_k = 0$. Such a choice reduces (5.1) to the well-known l_1 merit function and our algorithm becomes very similar to that analysed in [Han76]. When (5.2) is used, we assume conditions **MC1** and **MC3** hold. Again it is relatively trivial to satisfy these conditions. We have also assumed in the proofs that $\lambda_0 \ge 0$ and $\mu_k \ge 0$.

The criteria given in section 2 for the choice of steplength α_k assumes the merit function has continuous first derivatives. This property does not necessarily hold for the merit functions under consideration. Therefore we use the following criteria for determining a value α_k. The steplength α_k is required to satisfy

$$\phi(\alpha_k) \equiv \phi(x_k + \alpha_k p_k, \lambda_k + \alpha_k \xi_k) \le \phi(0) + \eta\alpha_k\Delta_k, \tag{5.3}$$

where $\Delta_k \equiv g_k^T p_k + (\xi_k - \lambda_k)^T c^-(x_k) - \rho_k\|c^-(x_k)\|$, and in addition must either satisfy

$$\alpha_k \ge \gamma_l > 0 \tag{5.4}$$

or

$$\alpha_k > \gamma_u \bar{\alpha}_k \qquad \text{and} \qquad \phi(\bar{\alpha}_k) > \phi(0) + \sigma\bar{\alpha}_k\Delta_k, \tag{5.5}$$

where $0 < \gamma_l < \gamma_u < 1$, $0 < \eta \le \sigma < 1$ and $\bar{\alpha}_k > 0$. For a discussion of this criteria and the existence of α_k see Calamai and Moré [CM87]. Our preference for the criteria given in section 2 is based on our belief that in practice they lead to a better choice of α_k. In the definition of our algorithm we could have used other steplength criteria without impacting the convergence properties.

The following basic relationships will be used to establish the convergence results,

$$c_i^-(x+\alpha p) \le |c_i(x+\alpha p) - c_i(x) - \alpha a_i^T p| + (c_i(x) + \alpha a_i^T p)^- \tag{5.6a}$$
$$(c_i(x) + \alpha a_i^T p)^- \le (1-\alpha)c_i^-(x) \tag{5.6b}$$
$$-\omega^T A p \le -\|c_i^-(x)\|, \tag{5.6c}$$

where $\omega^T A p$ represents an element of $\partial\varphi(0)$, the subdifferential of $\varphi(\alpha) \equiv \|c^-(x+\alpha p)\|$ at 0. The elements of ω take values in $[0,1]$ and are 0 whenever $c_i(x) > 0$. To simplify the notation in what follows, we introduce a diagonal matrix Ω formed from the elements of ω as $\Omega = \mathrm{diag}(\omega_i)$.

Consider now the case when ϕ has been defined from (5.1). From our assumption that $\lambda_k \geq 0$,

$$\lambda_k^T \Omega_k (A_k p_k + c_k) \geq 0$$

for all k. It follows from this inequality and the relationships given in (5.6) that

$$\phi_k'(0) = g_k^T p_k + \xi_k^T c^-(x_k) - \lambda_k^T \Omega_k A_k p_k - \rho_k \omega_k^T A_k p_k \leq \Delta_k.$$

We select ρ_k such that

$$\Delta_k \leq -\tfrac{1}{2} p_k^T H_k p_k. \tag{5.7}$$

This rule is analogous to the ones used in Byrd and Nocedal [BN88], and Burke and Han [BH89].

The first step is to establish that such a value of ρ exists. From Lemma 3.5 we have

$$\Delta_k \leq -\beta_1 \|p_k\|^2 + \beta_2 \|c_k^-\| - \tfrac{1}{2} p_k^T H_k p_k + \|\xi_k - \lambda_k\| \|c_k^-\| - \rho \|c_k^-\|,$$

and defining $\rho_u \equiv \beta_2 + 3\beta_{nmu}$, for any value $\rho \geq \rho_u$ condition (5.7) is satisfied for any k. This result also shows that the value of ρ will remain bounded in the algorithm.

Theorem 5.1. *The algorithm modified to use the merit functions (5.1) converges globally.*

Proof. Given the results from Section 3, it is enough to show that $\|p_k\| \to 0$.

As ρ cannot grow without bound, any strategy for increasing ρ by a finite quantity when required to do so must imply that there exists an iteration value K such that $\rho_k = \rho_K$ for all $k \geq K$. We consider only iterations of this form. For $k \geq K$,

$$\phi(\alpha_k) - \phi(\alpha_{k-1}) \leq \alpha_k \sigma \Delta_k \leq -\sigma \beta_{svH} \alpha_k \|p_k\|^2.$$

From the boundedness of ϕ, it follows that $\alpha_k \|p_k\|^2 \to 0$.

If $\|p_k\| \to 0$, convergence follows from Lemma 3.1. Otherwise, if for a subsequence $\|p_k\| > \epsilon$, we must have $\alpha_k \to 0$ along the subsequence, and from the termination conditions for the linesearch, $\bar{\alpha}_k \to 0$.

From the definition of the merit function (5.1), (we drop the index k in the following relationships)

$$\begin{aligned}\phi(\bar{\alpha}) - \phi(0) = {} & \bar{\alpha} g^T p + \lambda^T (c^-(\bar{\alpha}) - c^-(0)) + \bar{\alpha} \xi^T c^-(\bar{\alpha}) - \bar{\alpha} \rho \|c^-(0)\| \\ & + (F(\bar{\alpha}) - F(0) - \bar{\alpha} g^T p) + \rho(\|c^-(\bar{\alpha})\| - (1 - \bar{\alpha})\|c^-(0)\|).\end{aligned}$$

For the last term, from (5.6a) and (5.6b), it follows that

$$\|c^-(\bar{\alpha})\| - (1 - \bar{\alpha})\|c^-(0)\| \leq \|c(\bar{\alpha}) - c(0) - \bar{\alpha} A p\|,$$

and from this result we must have

$$\phi(\bar\alpha) - \phi(0) \le \bar\alpha g^T p + \lambda^T(c^-(\bar\alpha) - c^-(0)) + \bar\alpha \xi^T c^-(\bar\alpha) - \bar\alpha\rho\|c^-(0)\| \\ + (F(\bar\alpha) - F(0) - \bar\alpha g^T p) + \rho\|c(\bar\alpha) - c(0) - \bar\alpha A p\|.$$

If we use again (5.6a) and (5.6b) on the terms associated with the multiplier estimates (given that by assumption $\lambda + \bar\alpha\xi \ge 0$), we obtain

$$\phi(\bar\alpha) - \phi(0) \le \bar\alpha g^T p + \textstyle\sum_i (\lambda_i + \bar\alpha\xi_i)^T |c_i(\bar\alpha) - c_i(0) - \bar\alpha a_i^T p| + (1-\bar\alpha)\lambda^T c^-(0) \\ - \lambda^T c^-(0) + \bar\alpha(1-\bar\alpha)\xi^T c^-(0) - \bar\alpha\rho\|c^-(0)\| + O(\|\bar\alpha p\|^2).$$

Simplifying this expression gives

$$\phi(\bar\alpha) - \phi(0) \le \bar\alpha\Big(g^T p + (\xi - \lambda)^T c^-(0) - \rho\|c^-(0)\|\Big) + \bar\alpha^2 \|c^-(0)\|\,\|\xi\| + O(\|\bar\alpha p\|^2). \tag{5.8}$$

Replacing this bound in the linesearch termination condition implies

$$0 < (1-\sigma)\bar\alpha\Delta + \bar\alpha^2\|c^-(0)\|\,\|\xi\| + O(\|\bar\alpha p\|^2). \tag{5.9}$$

If in this equation we use the bound $\Delta \le -\beta_{svH}\|p\|^2$ and we take into account that $\|p\|$ is bounded away from zero, by taking limits along the subsequence we obtain

$$0 \le -(1-\sigma)\beta_{lvH}\epsilon^2,$$

but this is not possible, so we must have $\|p_k\| \to 0$ for the whole sequence. ∎

We now consider the second merit function (5.2). The subgradient along the search direction at (x_k, λ_k) is given by

$$\phi_k'(0) = g_k^T p_k + \xi_k^T c^-(x_k) - \lambda_k^T \Omega_k A_k p_k - \rho_k c^-(x_k)^T A_k p_k \le \Delta_k,$$

where we have defined

$$\Delta_k \equiv g_k^T p_k + (\xi_k - \lambda_k)^T c^-(x_k) - \rho_k \|c^-(x_k)\|^2,$$

and we have made use of $\lambda_k \ge 0$ to have

$$(\Omega_k\lambda_k + \rho_k c_k^-)^T(A_k p_k + c_k) \ge 0.$$

In this case it is not immediately evident that ρ_k remains bounded. The convergence proof we give is similar to the one introduced in Section 3. The definition of ρ given in that section will be preserved, replacing only $c - s$ by c^-.

Theorem 5.2. *The algorithm modified to use the merit function (5.2) converges globally.*

Proof. Again, from the results in Section 3 it is enough to show that $\|p_k\| \to 0$, although now we need to consider two cases.

If ρ_k remains bounded, then as in the proof of Theorem 5.1, we must have that $\alpha_k\|p_k\|^2 \to 0$.

If $\|p_k\| \to 0$, convergence follows from Lemma 3.1. Otherwise, if for a subsequence $\|p_k\| > \epsilon$, we must have $\alpha_k \to 0$, and from the termination conditions for the linesearch, $\bar{\alpha}_k \to 0$.

The argument when ρ is bounded is similar to that given in Theorem 5.1. From (5.6a) and (5.6b), we also have (we drop the index k in the following relationships)

$$\begin{aligned}\phi(\bar{\alpha}) - \phi(0) \le\ & \bar{\alpha} g^T p + \lambda^T(c^-(\bar{\alpha}) - c^-(0)) + \bar{\alpha}\xi^T c^-(\bar{\alpha}) - \rho(\bar{\alpha} - \tfrac{1}{2}\bar{\alpha}^2)\|c^-(0)\|^2 \\ & + \rho\|c(\bar{\alpha}) - c(0) - \bar{\alpha}Ap\|\left(\tfrac{1}{2}\|c(\bar{\alpha}) - c(0) - \bar{\alpha}Ap\| + \|(c(0) + \bar{\alpha}Ap)^-\|\right) \\ & + (F(\bar{\alpha}) - F(0) - \bar{\alpha} g^T p),\end{aligned}$$

and again using (5.6a) and (5.6b) on the terms associated to the multiplier estimates, we obtain

$$\begin{aligned}\phi(\bar{\alpha}) - \phi(0) \le\ & \bar{\alpha}\left(g^T p + (\xi - \lambda)^T c^-(0) - \rho\|c^-(0)\|^2\right) \\ & + \bar{\alpha}^2\|c^-(0)\|\left(\|\xi\| + \tfrac{1}{2}\rho\|c^-(0)\|\right) + O(\|\bar{\alpha}p\|^2).\end{aligned} \tag{5.10}$$

Replacing this bound in the linesearch termination condition implies

$$0 < (1-\sigma)\bar{\alpha}\Delta + \bar{\alpha}^2\|c^-(0)\|\left(\|\xi\| + \tfrac{1}{2}\rho\|c^-(0)\|\right) + O(\|\bar{\alpha}p\|^2). \tag{5.11}$$

If in this equation we use the bound $\Delta \le -\beta_{svH}\|p\|^2$ and we take into account that $\|p\|$ is bounded away from zero and ρ is bounded, by taking limits along the subsequence we obtain

$$0 \le -(1-\sigma)\beta_{lvH}\epsilon^2,$$

but this is not possible, so we must have $\|p_k\| \to 0$ for the whole sequence.

The other possible situation corresponds to ρ_k growing without bound. In this case we have that for all iterations where the value of the penalty parameter is increased

$$\rho_{k_l}\|c^-_{k_l}\| \le K_1 \qquad \text{and} \qquad \rho_{k_l}\|p_{k_l}\|^2 \le K_2.$$

The proof of this result is basically that of Lemma 3.7. From these bounds it is possible to show that we must also have

$$\rho_k\|p_k\|^2 \le K$$

for all k (the proof is similar to the one for Lemma 4.6 in [GMSW86b], or the one for Lemma 3.6.3 in [Pr89]), implying $p_k \to 0$ and the convergence of the algorithm. ∎

Near the solution we have shown a minimizer of the QP subproblem is determined. Consequently, the issue of superlinear convergence is not relevant to whether or not an incomplete SQP method is used (once convergence has been established). In some cases the algorithms described in this section (for specific choices of multiplier estimates) are similar to known algorithms for which superlinear convergence is not always attained without some modifications. For example, it has been demonstrated that the algorithm described in [Han76] may not take a unit step even when the sequence $\{x_k + p_k\}$ is superlinearly convergent (as our algorithm does when using our original merit function).

6. Numerical Results

In this section we present numerical results obtained from an implementation of our algorithm. We are unaware of any general purpose large-scale SQP routine on which we can test our modifications. As a first step we have modified the code NPSOL, which is an implementation of the algorithm NPSQP. We have called the modified routine INPSOL. Apart from the definition of the search direction all other aspects of INPSOL are identical to those of NPSOL. A detailed description of NPSOL is given in Gill *et al.* [GMSW86a]. It should be noted that NPSOL does not incorporate linear constraints into the merit function. An initial point is obtained that is feasible with respect to the linear constraints and thereafter feasibility is retained (by incorporating the linear constraints in the QP subproblem). On many practical problems the feasible region with respect to the linear constraints is compact. On such problems this approach ensures assumption **A1** is satisfied.

The purpose of the testing reported is to demonstrate that the efficiency and robustness of the modified algorithm are comparable to those of NPSOL. Naturally, we can only test the hypothesis on the domain of problems NPSOL is designed to solve, namely problems having a small number of variables and constraints, although on these problems the opportunities for improvement are limited, as we discuss later. What this implementation really tests is whether the introduction of flexibility in the determination of the search direction has a significant cost.

The search direction

The algorithm described in Section 2 allows for considerable flexibility of design. We describe here the specific choices made in our implementation. The search direction p_k is computed according to the following steps. (The subscript k corresponding to the iteration number is dropped from now on.)

- An initial feasible point p^0 is obtained following the same procedure as NPSOL. No special attempt was made to satisfy condition (2.7) since on the problems tested it was always satisfied by the feasibility phase in NPSOL.

- The active-set method used in NPSOL was terminated at $\hat{p}$, the *first* stationary point. At $\hat{p}$ the corresponding QP multipliers $\tilde{\mu}$ are computed. Define $\hat{\mu}$ as

$$\hat{\mu}_i \equiv \tilde{\mu}_i \|a_i\|.$$

- Let ϵ_M denote machine precision. If

$$\forall i \quad \hat{\mu}_i \geq -\sqrt{\epsilon_M}, \tag{6.1}$$

then $\hat{p}$ is taken as the search direction.

- If (6.1) does not hold, we can take a step away from a subset of the active constraints while decreasing the value of the QP objective function. To identify the set of active constraints to be deleted, define

$$\mu_{\min} \equiv \min_i \hat{\mu}_i,$$

and introduce a vector e_I as

$$(e_I)_i \equiv \begin{cases} \|a_i\| & \text{if } \hat{\mu}_i \leq 10^{-3}\mu_{\min}, \\ 0 & \text{otherwise.} \end{cases} \tag{6.2}$$

- There is also a limit of 50 on the maximum number of constraints to be deleted. If (6.2) is satisfied by more than 50 active constraints, only the ones having the smallest multipliers are deleted. For most problems this limit has no effect, since the total number of constraints is less than 50.

- The direction away from the selected constraints is obtained as the least-norm solution of the system

$$Ad = e_I;$$

that is, we define

$$d_Y = (AY)^{-1}e_I, \qquad d_Z = 0,$$

to obtain

$$d = Yd_Y.$$

- If α_c denotes the step to the nearest inactive constraint, and α_m is defined as in (2.9),

$$\alpha_m = -\frac{(g + H\hat{p})^T d}{d^T H d},$$

we define α as in (2.9):

$$\alpha = \min(\alpha_c, \alpha_m, \alpha_M),$$

where α_M is 10^{10} for this case.

- We obtain the search direction p from (2.10), as

$$p \equiv \begin{cases} \hat{p} + \alpha d & \text{if } \|\hat{p}\| < \beta_{slp}\|\hat{p} + \alpha d\|, \\ \hat{p} & \text{otherwise,} \end{cases}$$

where $\beta_{slp} = 100$; with this value the step αd is accepted in nearly all cases.

- Finally, the multiplier estimate used to define the linesearch is taken to be $\tilde{\mu}$ if $p = \hat{p}$. Otherwise, it is taken to be the least-squares estimate λ_L obtained from

$$AA^T\lambda_L = Ag.$$

Test problems

The two algorithms, NPSOL and INPSOL, have been compared by solving a collection of 114 problems from the literature.

The problems have been obtained from the following sources:

- Problem 1 is the example problem distributed with NPSOL; its description can be found in [GMSW86a]. Problems 3 and 4 are slight reformulations of the same problem, where the bounds $-1 \le x_3 \le 1$ have been replaced by the constraint $x_3^2 \le 1$. Problem 3 uses the starting point

$$\left(\tfrac{1}{3}, \tfrac{2}{3}, \tfrac{11}{10}, \tfrac{2}{3}, \tfrac{1}{3}, \tfrac{1}{3}, \tfrac{2}{3}, -\tfrac{1}{3}, -\tfrac{1}{3}\right).$$

- Descriptions for problems 6 and 12–15 can be found in [MS82]. The version of problem 6 considered is the one corresponding to a value $T = 10$. Problems 12 and 13 start from point (d) for Wright No. 4 as indicated in the reference, while problems 14 and 15 start from points (a) and (b) for Wright No. 9, respectively.
- A description of the SQUARE ROOT problems (17–20) and of EXP6 (9) can be found in Fraley [Fra88].
- Problems 21–30 were obtained from Boggs and Tolle [BT84].
- All problems having names starting with "HS" are from Hock and Schittkowski [HS81].
- Problems 85–95 can be found in Dembo [Dem76].

All the above problems have been used in the past to test NPSOL. It should be noted that the problems in this group are small; the average number of variables is 10, and the average number of constraints is 6. Nevertheless, many of these problems are considered hard to solve. Moreover, for some of these problems the assumptions made to establish the convergence results fail to hold; for example, in some cases the Jacobian at the solution is singular, or no feasible points exist for some QP subproblems.

The algorithms have also been tested on another group of problems.

- The structural optimization problems 99–114 are described in Ringertz [Rin88]. The letters "I" and "E" in the problem name indicate if the formulation used included explicitly the displacement variables ("E") or eliminated them in advance. Also, the following number (10, 25, 36 or 63) denotes the number of bars in the truss considered. Finally, whenever a number is included at the end of the name (006, 040 or 060), the initial point taken has been modified to be $x_j = 6$, 40 or 60 respectively.

These problems have been introduced due to the atypical behavior of quasi-Newton SQP algorithms on them. For this group, the ratio of QP to nonlinear iterations is large when compared to the size of the problem; on the first test set (problems 1–98) the average ratio for NPSOL is 2 QP iterations per nonlinear iteration, while on problems 99–114 the average ratio is 30.

The normal behavior of NPSOL on the first set of test problems is to require a relatively large number of QP iterations in the first few nonlinear iterations. Typically, the number of QP iterations declines exponentially until near the solution, when only one iteration is required. The STRUC problems depart from this "standard" behavior, in the sense that the number of QP iterations declines much more gradually. (Although only one QP iteration is required in the end, most nonlinear iterations require more.) This offers the possibility of observing the reductions that can be achieved by using the early-termination criterion, with limited distortion from the asymptotic behavior of NPSOL.

Finally, the problems in this second group are larger than the ones presented above; the average number of variables is now 55, and the average number of constraints is 100. For all the reasons mentioned, this set of problems provides a better environment in which to test the ability of the proposed early-termination criterion to reduce the number of QP iterations.

Computing environment

Version 4.02 of NPSOL was used in these comparisons. For this test set, all parameters used in the code have been fixed at their default values (see [GMSW86a]). No attempt was made to improve the results by selecting a different set of parameters. It would be difficult to compare the relative effort to adjust input parameters for the two algorithms. The runs were performed as batch jobs on a DEC VAXstation II with 5 Mb main memory. The operating system was VAX/VMS version 4.5, and the compiler used was VAX FORTRAN version 4.6 with default options.

Results

The results obtained from running both algorithms on the test set are presented in Table 2.

The parameters chosen to characterize the relative performance of both algorithms have been: the number of outer (nonlinear) iterations for each problem; the number of calls to the routine computing the values of the objective function, the constraint functions and their derivatives (function evaluations); the total number of inner (QP) iterations for the problem (this includes the number of iterations necessary to compute a feasible point); and the running (CPU) time needed to solve the problem. The results corresponding to both algorithms are given as a single entry in the tables, with the figures separated by a "/" symbol, in the form

NPSOL result/INPSOL result.

Given that many of the problems are not convex, the algorithms may converge to different solutions. A few such events are indicated in Table 2. Another possible outcome is failure—that is, the algorithm terminates without finding a solution, because the iteration limit has been exceeded, because no significant progress can be made at the current point with respect to the merit function, or because the objective or constraint functions need to be evaluated at a point for which they are not defined in the code. Such failures are indicated by "—".

For the set of 114 problems, NPSOL was able to find a solution in 107 cases, while INPSOL was able to solve 105 problems. We should emphasize that only the default value of the input parameters were used. Undoubtedly adjustment of the input parameters on the problems that failed would have led to more successes. The figures illustrate the reliability of INPSOL.

Table 1 presents a summary of the results for the four quantities monitored in Table 2. The average values have been computed as the geometric means for the ratios of the values for NPSOL and for INPSOL; that is, averages larger than one indicate that the corresponding value for NPSOL is larger than the value for INPSOL. Also, the averages exclude those problems where one of the algorithms failed. Separate entries have been provided for problems 1–98 (the smaller problems), and for problems 99–114 (the structural optimization problems).

TABLE 1
Average Behavior: NPSOL vs. INPSOL

	Problems		
	All	1–98	99–114
Nonlinear iterations	.988	.979	1.044
Function evaluations	.994	.999	.963
QP iterations	1.190	1.112	1.884
CPU time	1.043	1.022	1.200

We now comment briefly on the implications of these results.

- The early-termination rule seems to behave very well regarding the numbers of nonlinear iterations and function evaluations; even if we are now using a search direction of "worse quality" than in NPSOL, the numbers are very close for both algorithms.

- The number of QP iterations is reduced by 20% for the complete set. When judging this figure we must take into account that the problems are small, implying that the number of QP iterations required per nonlinear iteration is also small. (In fact, the average value for the test set is 5.6 QP iterations per nonlinear iteration.) The opportunity for improvement is correspondingly limited. Moreover, both codes use the active set at the solution of the previous QP subproblem as a prediction for the correct active set in the current subproblem, resulting in a small number of QP iterations close to the solution. As a result, significant savings achieved by incomplete solution of QP subproblems in the early iterations are masked by a large number of subproblems requiring only a few QP iterations. As an example, for problem 98 the largest number of QP iterations needed in any nonlinear iteration is reduced from 57 for NPSOL to 15 for INPSOL. This effect is much less clear when we look at total numbers of QP iterations (244 for NPSOL vs. 170 for INPSOL). Recall that it is necessary in any implementation to limit the number of iterations taken to solve the subproblem. This large reduction in the maximum number of iterations is encouraging. Moreover, it indicates that INPSOL and NPSOL took quite different paths to obtain a solution on many of the problems. In the light of this fact the similarity of performance is quite remarkable. Finally, the early-termination rule still requires a feasible point, and the feasibility phase is the same as in NPSOL. When this phase accounts for most of the total number of iterations, as with the STRUC problems, the possibility of improvement is further diminished.

Nonetheless, it should be noted that for problems 99–114 the improvement obtained is significantly greater than 20%, as the mean ratio is now 1.88; in fact, when we look only at the larger problems, the relative performance of INPSOL improves markedly. This offers the promise that for even larger problems the results obtained may be substantially better than the values shown above.

- The CPU time required by INPSOL is lower than the time for NPSOL, but by a factor that is much smaller than for the number of QP iterations. This is due not only to the fact that function evaluations can be expensive when compared to the effort to solve each QP subproblem, but also to some details in the implementation that have been chosen to affect the number of QP iterations, even at the expense of running time. For example, the multiplier estimate used for the linesearch (the least-squares multiplier) is expensive to compute when many constraints are deleted in the last step, as the factorization for the Jacobian of the active constraints must be updated. There are still options to be explored that might reduce the CPU time for the modified algorithm.

TABLE 2
Numerical Results

No.	Problem name	Nonlinear iterations	Function evaluations	QP iterations	CPU time (s)
1	NPSOL SAMPLE PROBLEM	12/13	16/18	45/34	3.69/3.61
2	SINGULAR	15/15	16/16	4/4	1.03/1.05
3	HEXAGON	15/16	21/23	32/29	4.41/4.41
4	HEXAGON (ALT. START)	11/11	16/14	35/26	3.56/3.26
5	LC7	7/9	9/11	13/16	.76/.95
6	ALAN MANNE'S PROBLEM	17/17	18/18	40/37	21.13/21.92
7	ROSEN-SUZUKI	8/8	11/11	9/9	.81/.81
8	QP PROBLEM	8/10	9/11	23/15	1.10/1.04
9	EXP6	33/53	35/57	38/57	1.96/3.08
10	STEINKE2	—*/5	—/6	—/14	—/.87
11	NORWAY	4/6†	5/7	34/13	1.23/.65
12	MHW4	10/10	18/15	14/12	1.31/1.25
13	MHW9	30/19†	56/28	42/24	3.71/2.31
14	MHW9 INEQUALITY 1	28/23	38/28	59/40	3.41/2.73
15	MHW9 INEQUALITY 2	41/14†	58/27	80/24	4.83/1.77
16	WOPLANT	25/29	29/33	44/35	6.85/7.17
17	SQUARE ROOT 1	—*/—*	—/—	—/—	—/—
18	SQUARE ROOT 2	23/23	36/36	0/0	5.01/5.32
19	SQUARE ROOT 3	6/6	9/9	7/7	.95/.94
20	SQUARE ROOT 4	—*/—*	—/—	—/—	—/—
21	BT1	11/11	19/19	11/11	.81/.83
22	BT2	9/9	14/14	9/9	.71/.70
23	BT3	2/2	5/5	2/2	.19/.19
24	BT4	12/12	18/18	13/13	.92/.92
25	BT5-HS63	6/6	9/9	8/8	.58/.58
26	BT6-HS77	15/15	21/21	16/16	1.52/1.54
27	BT7	31/31	56/56	32/32	3.36/3.43
28	BT8	17/17	19/19	17/17	1.25/1.44
29	BT9-HS39	13/13	16/16	14/14	.95/1.19
30	BT10	8/8	11/11	0/0	.48/.52
31	BT11-HS79	9/9	12/12	10/10	1.05/1.06
32	BT12	27/27	57/57	28/28	3.04/3.04
33	BT13	32/32	44/44	34/34	2.61/2.62
34	POWELL TRIANGLES	23/15	37/16	36/23	3.27/2.28
35	POWELL BADLY SCALED	12/12	15/15	13/13	.85/.85
36	POWELL WRIGGLE	34/32	69/55	60/40	2.77/2.39
37	POWELL-MARATOS	6/6	7/7	6/6	.44/.44
38	HS72	7/7	8/8	8/8	.69/.67
39	HS73 (CATTLE FEED)	4/4	5/5	4/4	.38/.36
40	HS107	11/11	18/18	27/18	2.77/2.56
41	MUKAI-POLAK	10/10	16/16	13/13	1.08/1.11
42	INFEASIBLE SUBPROBLEM	—*/—*	—/—	—/—	—/—
43	HS26	47/47	64/64	48/48	3.39/3.41
44	HS32	2/4	3/5	3/5	.25/.38
45	HS46	55/55	58/58	56/56	5.26/4.98
46	HS51	2/2	5/5	2/2	.18/.14
47	HS52	2/2	5/5	2/2	.19/.16
48	HS53	2/2	5/5	2/2	.19/.16
49	PENALTY1 A	16/16	18/19	77/41	20.01/16.49
50	PENALTY1 B	6/7	14/19	67/32	14.77/11.77
51	PENALTY1 C	29/15	85/40	152/65	24.35/11.65
52	HS13	22/19	23/20	13/10	1.29/1.22
53	HS64	29/43	39/62	47/60	2.34/3.33
54	HS65	8/9	10/11	16/16	.70/.78
55	HS70	36/—*	39/—	39/—	3.33/—
56	HS71	5/7	6/9	9/9	.53/.67
57	HS74	10/26	15/48	14/28	1.17/2.68

* Failed to solve the problem.
† Converged to a different minimizer.

TABLE 2 (CONT.)
Numerical results

No.	Problem name	Nonlinear iterations	Function evaluations	QP iterations	CPU time (s)
58	HS75	6/8	10/11	7/9	.72/.90
59	HS78	10/10	14/14	11/11	1.15/1.15
60	HS80	8/8	10/10	8/8	.92/.92
61	HS81	14/14	20/20	15/15	1.57/1.60
62	HS84	—*/4	—/5	—/9	—/.51
63	HS85	17/14	18/15	33/20	4.00/3.12
64	HS86 (COLVILLE 1)	6/7	8/8	11/11	.62/.64
65	HS87 (COLVILLE 6)	11/8	18/9	18/14	1.63/1.23
66	HS93	12/12	15/15	14/14	1.36/1.38
67	HS95	1/1	2/2	1/1	.15/.15
68	HS96	1/1	2/2	1/1	.17/.15
69	HS97	3/3	6/6	3/3	.40/.41
70	HS98	3/3	6/6	8/8	.43/.44
71	HS99	23/—*	44/—	74/—	3.99/—
72	HS100	14/14	29/29	18/18	2.07/2.02
73	HS104	18/18	20/20	23/23	3.36/3.37
74	HS105	43/—*	61/—	97/—	27.14/—
75	HS108 (HEXAGON)	24/32	45/49	57/87	6.78/9.36
76	HS109	11/10	13/11	25/29	3.23/3.26
77	HS110	6/6	9/9	24/15	.78/.69
78	HS111	41/49	64/75	44/52	8.08/9.05
79	HS112 (CHEMICAL EQ.)	19/—*	39/—	54/—	2.78/—
80	HS113	14/16	19/23	38/36	3.12/3.41
81	HS114	18/16	19/24	36/33	3.81/3.60
82	HS117 (COLVILLE 2)	17/18	21/27	96/39	6.75/5.34
83	HS118 (LC PROBLEM)	4/4	6/6	20/20	1.35/1.40
84	HS119 (COLVILLE 7)	12/17	16/19	41/47	4.25/5.60
85	DEMBO 1B	281/—*	437/—	296/—	75.46/—
86	DEMBO 2-HS83	4/4	6/6	4/4	.54/.54
87	DEMBO 3	9/8	11/9	37/20	2.01/1.78
88	DEMBO 4A	19/19	23/23	24/24	3.53/3.31
89	DEMBO 4C	13/13	15/15	20/23	3.10/3.20
90	DEMBO 5-HS106	17/18	21/24	30/31	2.90/3.04
91	DEMBO 6-HS116	36/43	96/69	144/248	21.84/29.65
92	DEMBO 7	19/12	24/15	126/68	15.54/9.82
93	DEMBO 8A	33/42	85/118	105/99	7.52/9.17
94	DEMBO 8B	29/29	69/71	88/73	6.51/6.45
95	DEMBO 8C	25/27	60/68	89/65	6.19/6.06
96	OPF	18/17	19/18	53/51	468.12/456.10
97	GBD EQUILIBRIUM MOD.	5/6	6/7	37/26	6.22/6.10
98	WEAPON ASSIGNMENT	96/73	98/76	244/170	120.78/114.93
99	STRUCI10KON	18/17	34/30	65/42	13.67/11.73
100	STRUCE10KON	26/29	49/67	87/84	17.68/20.75
101	STRUCI10VAN	23/19	41/34	54/51	16.30/13.85
102	STRUCE10VAN	—*/24	—/48	—/91	—/19.44
103	STRUCI25006	42/37	68/62	147/85	92.44/80.99
104	STRUCE25006	20/28	32/36	178/95	357.83/260.79
105	STRUCI25DAT	11/12	19/21	24/22	24.75/27.11
106	STRUCE25DAT	52/21	106/37	687/65	647.13/191.44
107	STRUCI36DAT	23/20	38/34	59/46	120.79/108.02
108	STRUCE36DAT	29/30	53/62	87/90	971.16/1021.87
109	STRUCI63040	117/112	211/202	6116/3091	8182.13/7159.03
110	STRUCE63040	375/—*	794/—	3545/—	77286.64/—
111	STRUCI63060	—*/98	—/244	—/3899	—/8281.02
112	STRUCE63060	63/115	150/316	6675/3407	25090.15/33228.42
113	STRUCI63DAT	246/136	354/412	9043/2060	12591.61/11424.54
114	STRUCE63DAT	52/72	86/145	8049/2858	41793.84/22740.66

* Failed to solve the problem.
† Converged to a different minimizer.

7. References

[Big72] M.C. Biggs (1972), Constrained minimization using recursive equality quadratic programming, in: F.A. Lootsma, Ed., *Numerical Methods for Nonlinear Optimization* (Academic Press, London/New York).

[BT84] P.T. Boggs and J.W. Tolle (1984), A family of descent functions for constrained optimization, *SIAM Journal on Numerical Analysis* **21** 1146–1161.

[BTW82] P.T. Boggs, J.W. Tolle and P. Wang (1982), On the local convergence of quasi-Newton methods for constrained optimization, *SIAM Journal on Control and Optimization* **20** 161–171.

[BH89] J.V. Burke and S-P. Han (1989), A robust sequential quadratic programming algorithm, *Mathematical Programming* **43** 277–303.

[BN88] R.H. Byrd and J. Nocedal (1988), An analysis of reduced Hessian methods for constrained optimization, Report CU-CS-398-88, Department of Computer Science, University of Colorado.

[CM87] P.H. Calamai and J.J. Moré (1987), Projected gradient methods for linearly constrained problems, *Mathematical Programming* **39** 93–116.

[CDT85] M.R. Celis, J.E. Dennis, Jr. and R.A. Tapia (1985), A trust region strategy for nonlinear equality constrained optimization, in: P.T. Boggs, R.H. Byrd and R.B. Schnabel, Eds., *Numerical Optimization 1984* (SIAM, Philadelphia).

[Dem76] R.S. Dembo (1976), A set of geometric programming test problems and their solutions, *Mathematical Programming* **10** 192–213.

[DT85] R.S. Dembo and U. Tulowitzki (1985), Sequential truncated quadratic programming methods, in: P.T. Boggs, R.H. Byrd and R.B. Schnabel, Eds., *Numerical Optimization 1984* (SIAM, Philadelphia).

[Fle70] R. Fletcher (1970), A class of methods for nonlinear programming with termination and convergence properties, in: J. Abadie, Ed., *Integer and Nonlinear Programming* (North Holland, Amsterdam).

[Fle85] R. Fletcher (1985), An ℓ_1 penalty method for nonlinear constraints, in: P.T. Boggs, R.H. Byrd and R.B. Schnabel, Eds., *Numerical Optimization 1984* (SIAM, Philadelphia).

[Fle87] R. Fletcher (1987), *Practical Methods of Optimization* (John Wiley and Sons, Chichester/New York/Brisbane/Toronto/Singapore).

[Fra88] C. Fraley (1988), Software performance on nonlinear least-squares problems, SOL Report 88-17, Department of Operations Research, Stanford University.

[GMSW86a] P.E. Gill, W. Murray, M.A. Saunders and M.H. Wright (1986), User's guide for NPSOL (Version 4.0): a FORTRAN package for nonlinear programming, Report SOL 86-2, Department of Operations Research, Stanford University.

[GMSW86b] P.E. Gill, W. Murray, M.A. Saunders and M.H. Wright (1986), Some theoretical properties of an augmented Lagrangian merit function, Report SOL 86-6R, Department of Operations Research, Stanford University.

[GMSW88] P.E. Gill, W. Murray, M.A. Saunders and M.H. Wright (1988), Inertia-controlling methods for quadratic programming, Report SOL 88-3, Department of Operations Research, Stanford University. (To appear in SIAM Review.)

[GMW81] P.E. Gill, W. Murray and M.H. Wright (1981), *Practical Optimization* (Academic Press, London/New York).

[Han76] S.-P. Han (1976), Superlinearly convergent variable metric algorithms for general nonlinear programming problems, *Mathematical Programming* **11** 263–282.

[HS81] W. Hock and K. Schittkowski (1981), Test examples for nonlinear programming, Lecture Notes in Economics and Mathematical Systems, Vol. 187 (Springer-Verlag, Berlin/Heidelberg/New York).

[Mu69] W. Murray (1969), An algorithm for constrained minimization, in: R. Fletcher, Ed., *Optimization* (Academic Press, London/New York).

[MS82] B.A. Murtagh and M.A. Saunders (1982), A projected Lagrangian algorithm and its implementation for sparse nonlinear constraints, *Mathematical Programming Study* **16** 84–117.

[Po78] M.J.D. Powell (1978), A fast algorithm for nonlinearly constrained calculations, in: O.L. Mangasarian, R.R. Meyer and S.M. Robinson, Eds., *Nonlinear Programming 3* (Academic Press, New York).

[Po87] M.J.D. Powell (1987), Methods for nonlinear constraints in optimization calculations, Proceedings of the 1986 IMA/SIAM Meeting on "The State of the Art in Numerical Analysis", Clarendon Press, Oxford.

[PY86] M.J.D. Powell and Y. Yuan (1986), A recursive quadratic programming algorithm that uses differentiable exact penalty functions, *Mathematical Programming* **35** 265–278.

[Pr89] F.J. Prieto (1989), Sequential quadratic programming algorithms for optimization, Report SOL 89-7, Department of Operations Research, Stanford University.

[Rin88] U.T. Ringertz (1988), A mathematical programming approach to structural optimization, Report No. 88-24, Dept. of Aeronautical Structures and Materials, The Royal Institute of Technology, Stockholm.

[Rob74] S.M. Robinson (1974), Perturbed Kuhn-Tucker points and rates of convergence for a class of nonlinear programming algorithms, *Mathematical Programming* **7** 1–16.

[Sch81] K. Schittkowski (1981), The nonlinear programming method of Wilson, Han and Powell with an augmented Lagrangian line search function, *Numerische Mathematik* **38** 83–114.

[Wil63] R.B. Wilson (1963), A simplicial algorithm for concave programming, Ph.D. Thesis, Harvard University.

[Wri76] M.H. Wright (1976), Numerical methods for nonlinearly constrained optimization, Ph.D. Thesis, Stanford University.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	September 1990	Technical Report

4. TITLE AND SUBTITLE

A Sequential Quadratic Programming Algorithm Using an Incomplete Solution of the Subproblem

5. FUNDING NUMBERS

Grant N00014-90-J-1242

6. AUTHOR(S)

Walter Murray and Francisco J. Prieto

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Department of Operations Research - SOL
Stanford University
Stanford, CA 94305-4022

8. PERFORMING ORGANIZATION REPORT NUMBER

1111MA

9. SPONSORING MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Office of Naval Research - Department of the Navy
800 N. Quincy Street
Arlington, VA 22217

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

SOL 90-12

12a. DISTRIBUTION AVAILABILITY STATEMENT

UNLIMITED

12b. DISTRIBUTION CODE

UL

13 ABSTRACT *(Maximum 200 words)*

A feature of current sequential quadratic programming (SQP) methods to solve nonlinear constrained optimization problems is the necessity at each iteration to solve a quadratic program (QP). We show that if the QP subproblem is convex and an active-set method is used to solve it, then there exist iterates other than the minimizer that may be used to define a suitable search direction. None of the usual properties of an SQP method are compromised by the new definition of the search direction.

We derive some new properties for an SQP method that uses a particular augmented Lagrangian merit function. Specifically we show, under suitable additional assumptions, that the rate of convergence is superlinear. We also show that the penalty parameter used in the merit function is bounded.

14. SUBJECT TERMS

quasi-Newton; nonlinear programming; sequential quadratic programming.

15. NUMBER OF PAGES

38 pp.

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED			SAR